Alexander Gelbukh
Eduardo F. Morales (Eds.)

MICAI 2008: Advances in Artificial Intelligence

LNAI 5317

7th Mexican International Conference on Artificial Intelligence
Atizapán de Zaragoza, Mexico, October 2008
Proceedings

Springer

Lecture Notes in Artificial Intelligence 5317

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Alexander Gelbukh Eduardo F. Morales (Eds.)

MICAI 2008: Advances in Artificial Intelligence

7th Mexican International Conference
on Artificial Intelligence
Atizapán de Zaragoza, Mexico, October 27-31, 2008
Proceedings

Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Alexander Gelbukh
Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Laboratorio de Lenguaje Natural y Procesamiento de Texto
Av. Juan Dios Bátiz s/n, Nueva Industrial Vallejo, 07738 Mexico DF, Mexico
and
University of Central Europe (UCE)
Social Networks Research Center (SoNet RC)
Královská 386/11, 90901 Skalica, Slovakia
E-mail: gelbukh@gelbukh.com

Eduardo F. Morales
Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)
Ciencias Computacionales
Luis Enrique Erro 1, Santa María Tonantzintla, 72840 Puebla, Mexico
E-mail: emorales@inaoep.mx

Library of Congress Control Number: 2008936816

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-88635-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-88635-8 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12546381 06/3180 5 4 3 2 1 0

Preface

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. In 2008 Mexico celebrates the 50th anniversary of development of computer science in the country: in 1958 the first computer was installed at the National Autonomous University of Mexico (UNAM). Nowadays, computer science is the country's fastest growing research area.

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, 4293, and 4827. Since its foundation in 2000, the conference has been growing in popularity, and improving in quality.

This volume contains the papers presented at the oral session of the 7th Mexican International Conference on Artificial Intelligence, MICAI 2008, held October 27–31, 2008, in Atizapán de Zaragoza, Mexico. The conference received for evaluation 363 submissions by 1,032 authors from 43 countries (see Tables 1 and 2). This volume contains revised versions of 94 papers by 308 authors from 28 countries selected according to the results of an international reviewing process. Thus the acceptance rate was 25.9%. The book is structured into 20 thematic fields representative of the main current areas of interest for the AI community, plus a section of invited papers:

- Invited Papers
- Logic and Reasoning
- Knowledge-Based Systems, Knowledge Representation and Acquisition
- Ontologies
- Natural Language Processing
- Machine Learning
- Pattern Recognition
- Data Mining
- Neural Networks
- Genetic Algorithms
- Hybrid Intelligent Systems
- Computer Vision and Image Processing
- Robotics
- Planning and Scheduling
- Uncertainty and Probabilistic Reasoning
- Fuzzy Logic
- Intelligent Tutoring Systems
- Multi-agent Systems and Distributed AI
- Intelligent Organizations
- Bioinformatics and Medical Applications
- Applications

The conference featured excellent keynote lectures by leading AI experts:

- Gerardo Jiménez Sánchez, Mexico
- Stephanie Forrest, USA
- Francisco Cervantes Peréz, México
- Simon Haykin, Canada
- Steven M. LaValle, USA
- Georg Gottlob, UK / Austria

Table 1. Statistics of submissions (S), accepted (A) papers,[1] and acceptance rate (R) by country

Country or region	Authors		Papers		
	S	A	S	A	R
Algeria	9	–	4.33	–	–
Argentina	23	7	5.33	1	0.19
Australia	2	–	1.33	–	–
Austria	2	–	1	–	–
Belgium	2	2	0.33	0.33	1
Bolivia	1	–	0.12	–	–
Brazil	51	17	20.33	6.33	0.31
Canada	12	7	5.67	2.67	0.47
Chile	7	3	3.67	1.67	0.45
China	46	4	17.92	1	0.06
Colombia	7	2	4.25	1	0.24
Croatia	2	–	1	–	–
Cuba	25	13	8.75	3.67	0.42
Denmark	1	–	1	–	–
France	28	15	10.83	4.17	0.38
Greece	5	–	1	–	–
Hungary	2	2	0.33	0.33	1
India	27	5	8	1	0.12
Iran	18	4	7	1	0.14
Israel	4	2	2	1	0.5
Italy	10	7	2.58	2	0.77
Japan	14	–	4.48	–	–
Korea, South	14	–	6	–	–
Libya	1	–	1	–	–
Lithuania	3	1	2	1	0.5
Malaysia	9	2	3	1	0.33
Mexico	526	141	177.79	41.55	0.23
The Netherlands	3	3	1	1	1
Nicaragua	1	–	0.5	–	–
Pakistan	3	–	1	–	–
Poland	12	5	3.6	1.6	0.44
Portugal	6	2	3	1	0.33
Russia	1	1	0.2	0.2	1
Singapore	3	3	1	1	1
Slovakia	1	–	0.5	–	–
Spain	72	26	20.33	6.83	0.34
Sweden	4	3	1.5	1	0.67
Switzerland	1	–	0.2	–	–
Taiwan	14	6	6	2	0.33
Tunisia	3	–	1.33	–	–
Turkey	10	7	3	2	0.67
UK	12	7	5.92	2.75	0.46
USA	35	11	12.85	3.9	0.3
43	1032	308	363	65	0.25

[1] Counted by authors: e.g., for a paper by 2 authors from UK and 1 from USA, we added ⅔ to UK and ⅓ to USA.

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, a Doctoral Consortium, the First Mexican Symposium on Artificial Intelligence (COMIA), and a poster session, which were published in separate proceedings volumes.

The following papers received the Best Paper Award and the Best Student Paper Award:[1]

1st place: "A Knowledge Discovery Framework for Learning Task Models from User Interactions in Intelligent Tutoring Systems," by Philippe Fournier-Viger, Roger Nkambou, and Engelbert Mephu Nguifo, Canada / France

2nd place: "Effect of Preprocessing on Extractive Summarization with Maximal Frequent Sequences," by Yulia Ledeneva, Mexico

3rd place: "A Set of Test Cases for Performance Measures in Multiobjective Optimization," by Giovanni Lizárraga, Arturo Hernández, and Salvador Botello, Mexico

Student: "Spanish Nested Named Entity Recognition Using a Syntax-Dependent Tree Traversal-Based Strategy," by Yunior Ramírez-Cruz and Aurora Pons-Porrata, Cuba

[1] The best student paper was selected out of papers whose first author was a full-time student.

Table 2. Statistics of submissions and accepted papers by area

Accepted	Submitted	Rate	Area
10	34	0.29	Multi-agent Systems and Distributed AI
10	27	0.37	Natural Language Processing
9	36	0.25	Genetic Algorithms
6	29	0.20	Computer Vision and Image Processing
6	13	0.46	Hybrid Intelligent Systems
5	29	0.17	Applications
5	23	0.21	Machine Learning
5	17	0.29	Bioinformatics and Medical Applications
5	11	0.45	Intelligent Tutoring Systems
4	16	0.25	Pattern Recognition
4	16	0.25	Robotics
4	15	0.27	Logic and Reasoning
4	12	0.33	Fuzzy Logic
4	12	0.33	Planning and Scheduling
3	14	0.21	Neural Networks
2	9	0.22	Data Mining
2	8	0.25	Ontologies
2	3	0.66	Intelligent Organizations
2	3	0.66	Uncertainty and Probabilistic Reasoning
1	14	0.07	Expert Systems and Knowledge-Based Systems
1	8	0.12	Knowledge Representation and Management
–	7	–	Philosophical and Methodological Issues of AI
–	2	–	Spatial and Temporal Reasoning
–	1	–	Knowledge Acquisition
–	1	–	Case-Based Reasoning
–	1	–	Intelligent Interfaces: Multimedia, Virtual Reality
–	1	–	Qualitative Reasoning
–	1	–	Model-Based Reasoning
95	363		

We thank all the people involved in the organization of this conference. In the first place, the authors of the papers constituting this book: it is the excellence of their research work that gives value to the book. 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 Ángel Kuri and to Alejandro Peña Ayala for his help in the reviewing process.

We would like to express our sincere gratitude to the Tecnológico de Monterrey, Campus Estado de México for the warm hospitality to all MICAI 2008 attendees. Special thanks go to the Vice President of Rectoría Zona Centro, Dr. Roberto Rueda Ochoa, and to the Director General of Campus Estado de México, Dr. Pedro Grasa Soler, both at Tecnológico de Monterrey, for their valuable participation and support for the organization of this conference. Our gratitude also goes to the Division of Research and Graduate Studies and the Division of Engineering and Architecture at

Campus Estado de México of the same institute. We are grateful to the Conference Staff and Local Organizing Committee, led by Neil Hernández and Raúl Monroy.

Nothing would have been possible without the financial support of our sponsors. Special thanks go to the Dean of State of Mexico's Council for Science and Technology (COMECyT), Dr. Elías Micha, for financial support.

The entire submission and reviewing process, as well as putting together the proceedings, was supported by the EasyChair system (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 Springer staff's great patience and help in editing this volume.

August 2008

Alexander Gelbukh
Eduardo F. Morales

Organization

MICAI 2008 was organized by the Mexican Society for Artificial Intelligence (SMIA) in collaboration with Tecnológico de Monterrey, Campus Estado de México.

Conference Committee

General Chairs	Ángel Kuri Morales
	Carlos Alberto Reyes-Garcia
Program Chairs	Alexander Gelbukh
	Eduardo F. Morales
Workshop Chair	Grigori Sidorov
Tutorial Chair	Jesús González
Keynote Speaker Chairs	Raul Monroy
	Jesús González
AI Symposium Chair	José Galaviz Casas
Finance Chair	Grigori Sidorov
Logistics Chair	Raúl Monroy
Publicity Chair	Alejandro Peña
Grant Chair	Carlos Alberto Reyes-Garcia
Doctoral Consortium Chairs	Miguel González
	Manuel Montes

Program Committee

Chairs	Alexander Gelbukh
	Eduardo F. Morales

Each Program Committee member was Area Chair or Co-chair of one or more areas:

Ajith Abraham	Genetic Algorithms
	Hybrid Intelligent Systems
Alejandro Peña Ayala	Data Mining
	Applications
Alessandro Ricci	Multi-agent Systems and Distributed AI
	Intelligent Organizations
Alexander Gelbukh	Applications
	Philosophical and Methodological Issues of AI
Benjamin Kuipers	Qualitative Reasoning
	Model-Based Reasoning
Eduardo Morales	Uncertainty and Probabilistic Reasoning
Eugene Levner	Planning and Scheduling
Grigori Sidorov	Natural Language Processing

Ildar Batyrshin	Fuzzy Logic
	Spatial and Temporal Reasoning
Isaac Rudomin	Intelligent Interfaces: Multimedia, Virtual Reality
James Little	Computer Vision and Image Processing
Jose Ruiz-Shulcloper	Pattern Recognition
Juan J. Flores	Qualitative Reasoning
	Model-Based Reasoning
Juan-Manuel Serrano	Multi-agent Systems and Distributed AI
	Intelligent Organizations
Justin Dauwels	Neural Networks
Leif Peterson	Bioinformatics and Medical Applications
Lucia Maria Martins Giraffa	Intelligent Tutoring Systems
Luis Enrique Sucar	Computer Vision and Image Processing
Manuel Vilares Ferro	Natural Language Processing
Mario Koeppen	Genetic Algorithms
	Hybrid Intelligent Systems
Olac Fuentes	Machine Learning
	Data Mining
Rafael Murrieta	Robotics
Raul Monroy	Logic and Reasoning
Steve Legrand	Ontologies
Ulises Cortes	Expert Systems and Knowledge-Based Systems
	Knowledge Representation and Management
	Knowledge Acquisition
	Case-Based Reasoning
Zhe (Sage) Chen	Neural Networks

Technical Reviewing Committee

Andrea Schaerf
Abdelhamid Bouchachia
Adelmo Cechin
Adriano L. I. Oliveira
Aida Valls
Alberto Fernández
Alberto Taboada-Crispi
Alejandro Peña Ayala
Aleksander Valjamae
Alexandre Direne
Alexandru Floares
Alfredo Weitzenfeld
Alvaro Pardo
Ana Cerdeira-Pena
Ana-Luisa Solis
Andre Carvalho
Andre Raabe
Andrea Giovannucci
Andrea Schaerf
Andreas Nonnengart
Angel Kuri-Morales
Anna Maria Fanelli
Anne Denton
Annie Zaenen
Anton Bogdanovych
Antonio Carlos da Rocha Costa
Antonio Eleuteri
Antonio Fariña
Antonio Moreno
Arantza Aldea
Ariel Carrasco Ochoa
Arturo Hernandez-Aguirre
Augusto G. Boyano
Aurelio Lopez

Azzam Taktak
Beatriz López
Bedrich Benes
Berend Jan van der Zwaag
Bob Woodham
Candelaria E. Sansores
Carlos A. Iglesias
Carlos Alberto Reyes-Garcia
Carlos Carrascosa
Carlos Delgado-Mata
Carlos Mario Zapata Jaramillo
Carlos Merida-Campos
Carmelo Bastos-Filho
Carole Bernon
Christian Boitet
Christian Igel
Christian Veenhuis
Chuan Feng
Ciro Castiello
Claudia Esteves
Claudia P. Ayala
Claudia Zepeda
Claudio Toledo
Constantin Orasan
Corrado Mencar
Cristian Barrue
Daniel Acevedo
Daniel Ramirez
Daniela Zaharie
Danilo F. Carvalho
David Cabanillas
David Sanchez
Davide Grossi
Dekai Wu
Diana Francisca Adamatti
Dídac Busquets
Dieter Hutter
Diogo Pacheco
Domingo Mery
Edel Garcia
Edgar Alfredo Portilla-Flores
Edgar Chavez
Eduardo Alonso
Eduardo Garea
Efren Mezura-Montes
Emmanuel Hebrard
Enrico Pontelli
Enrique Marañón
Enrique Sucar
Enrique Vidal
Eric Manuel Rosales Peña Alfaro
Eric Matson
Estevam Hruschka Jr.
Ewa Szlachcic
Fabiola Lopez y Lopez
Faisal Momen
Federico Ambrogi
Federico Ramirez
Felipe Trevizan
Felix Calderon
Fernando B. Lima-Neto
Fernando Magan Muñoz
Francesco Masulli
Francisco Cantu
Francisco Carvalho
Francisco Estrada
Francisco Jose Ribadas Pena
Francois Modave
Francois Vialatte
Frank-Michael Schleif
Fuji Ren
Gabriel Acosta
Gang Xiang
Georg Gottlob
George Vouros
Gerald Schaefer
Gildardo Sanchez
Giovani Gomez Estrada
Graçaliz Dimuro
Graçaliz Pereira
Gregorio Toscano-Pulido
Gregory Randall
Guillermo Morales-Luna
Guillermo Sanchez
Gustavo Carneiro
Gustavo Olague
Hector Tejeda
Holger Billhardt
Homero Rios
Hongbo Liu
Horacio Rodriguez
Hugo Terashima
Humberto Sossa Azuela
Hyoil Han

Ildar Batyrshin
Ismael Lopez
Iván García-Magariño
Ivan Lopez-Arevalo
Jacak Witold
Jacek Malec
Jan Žižka
Jason Kessler
Javier Bejar
Javier Diez
Javier Flavio Vigueras Gomez
Javier Vazquez
Jean-Bernard Hayet
Jerzy Kotowski
Jesús Cérquides
Jesus Gonzalez
John Carroll
Jorge de la Calleja
Jose Calvo
Jose Fco. Martinez
Jose Luis Marroquin
Juan Antonio Navarro Perez
Juan Arturo Nolazco Flores
Juan Carlos Nieves
Juan Frausto Solis
Juan Gabriel Avina-Cervantes
Juan Jose Villanueva
Juan Lorenzo Ginori
Juan Otero Pombo
Juanjo Villanueva
Jürgen Dix
Justin Dauwels
Karina Gibert
Kenichi Mishina
Leliane Barros
Leonardo Garrido
Leonid Meyzin
Leonid Sheremetov
Leopoldo Altamirano
Louise Dennis
Lourdes Munoz-Gomez
Luciana Montera
Luisa Carpente
Luz Abril Torres-Méndez
Manuel Tupia
Marc Esteva
Marcello Balduccini
Marcia Moraes
Marco de Gemmis
Marco Pedersoli
Marc-Philippe Huget
Margarita Reyes-Sierra
Maria Alessandra Torsello
Maria Buemi
Maria del Pilar Gómez Gil
Marilton Aguiar
Mario Castelán
Mario Gutierrez
Mario Köppen
Mario Martin
Mark Kon
Marta Mejail
Masaki Murata
Matias Alvarado
Matjaž Gams
Mauricio Osorio
Michel Devy
Michele Piunti
Miguel Arias
Miguel González Mendoza
Leticia Maria Seijas
Lev Bregman
Miguel Leon
Mikhail Alexandrov
Milagros Fernández Gavilanes
Miquel Sànchez-Marrè
Miriam Eckert
Miroslav Bursa
Mohamed Abdel Fattah
Moises Alencastre-Miranda
Mostafa A. El-Hosseini
Nadezhda Yarushkina
Naveed Afzal
Nelly Koven
Nicandro Cruz
Nicolas Nicolov
Nicolas Sabouret
Nicolas Vandapel
Nicoletta Calzolari
Nicoletta Fornara
Nicomedes Cavalcanti Junior
Nikolay Shilov
Nils Bulling
Nuria Castell

Olivier Aycard
Pablo Arias
Pablo Dalbem de Castro
Pablo H. Ibarguengoytia
Pancho Tolchinsky
Patricia Jaques
Paulo Franca
Paulo Roberto Ferreira Jr.
Pavlos Peppas
Pedro Real Jurado
Peter Nightingale
Peter Novak
Philippe Pasquier
Phillip Burrell
Pilar Pozos Parra
Prabha Sharma
Prabu Dorairaj
Qiang Shen
Quim Comas
Rafael Falcon
Rafal Angryk
Ramon Baldrich
Ramon Sanguesa-Sole
Raul Monroy
Raul Sanchez
Renata Souza
René García Hernández
Reyes Rios-Cabrera
Ricardo Landa-Becerra
Ricardo Martins de Abreu Silva
Rimma Fatkullina
Robert Fisher
Roberto Araiza
Rocio Gonzalez
Rohit Parikh
Rosa Vicari
Ryszard Klempous
Salem Benferhat
Sandra Kuebler
Sergio Cano
Seth Hutchinson
SeungJin Lim
Shiyan Ou
Simon Colton
Son Tran
Stefanie Peters
Steliana Ivanova
Steven LaValle
Steven Willmott
Terence Etchells
Teresa Ludermir
Thamar Solorio
Thomas Villmann
Thomas Weise
Tianjing Feng
Tim Miller
Toby Walsh
Urszula Markowska-Kaczmar
Vaclav Snasel
Valerie Camps
Vicente Julian
Victor Ayala
Victor Manuel Darriba Bilbao
Victor Sosa
Viktor Pekar
Vladik Kreinovich
Xiao Sun
Yasunari Harada
Yulia Ledeneva

Organizing Committee

General Local Chairs	Raul Monroy Neil Hernandez
Planning Chair	Miguel González
Logistics Chair	Jorge Ramírez
Registration Chair	Maricela Quintana
Publicity Chair	Isabel Kreiner
Website Chair	Alejandro Aceves
Sponsor Chair	Ángeles Junco

Finance Chair	Raul Monroy
Webmasters	Ewin Cordoba
	Fernando Gudiño
General Support	Eber Villalobos

Webpage and Contact

The MICAI series website is www.MICAI.org. The website of the Mexican Society for Artificial Intelligence, SMIA, is www.SMIA.org.mx. Contact options and additional information can be found on these websites.

Table of Contents

Invited Papers

Logic and Reasoning

Knowledge-Based Systems, Knowledge Representation and Acquisition

Ontologies

Natural Language Processing

Machine Learning

Pattern Recognition

Data Mining

Neural Networks

Genetic Algorithms

Hybrid Intelligent Systems

Computer Vision and Image Processing

Robotics

Planning and Scheduling

Uncertainty and Probabilistic Reasoning

Fuzzy Logic

Intelligent Tutoring Systems

Multi-agent Systems and Distributed AI

Intelligent Organizations

Bioinformatics and Medical Applications

Applications

Heuristic Methods for Hypertree Decomposition

Artan Dermaku[1], Tobias Ganzow[2], Georg Gottlob[3,1,⋆],
Ben McMahan[4], Nysret Musliu[1], and Marko Samer[1]

[1] Institut für Informationssysteme (DBAI), TU Wien, Austria
[2] Mathematische Grundlagen der Informatik, RWTH Aachen, Germany
[3] Computing Laboratory, University of Oxford, UK
[4] Department of Computer Science, Rice University, USA

Abstract. The literature provides several structural decomposition methods for identifying tractable subclasses of the constraint satisfaction problem. *Generalized hypertree decomposition* is the most general of such decomposition methods. Although the relationship to other structural decomposition methods has been thoroughly investigated, only little research has been done on efficient algorithms for computing generalized hypertree decompositions. In this paper we propose new heuristic algorithms for the construction of generalized hypertree decompositions. We evaluate and compare our approaches experimentally on both industrial and academic benchmark instances. Our experiments show that our algorithms improve previous heuristic approaches for this problem significantly.

1 Introduction

Many important problems in artificial intelligence, database systems, and operations research can be formulated as *constraint satisfaction problems (CSPs)*. Such problems include scheduling, planning, configuration, diagnosis, machine vision, spatial and temporal reasoning, etc. [5]. A *CSP instance* consists of a finite set V of variables, a set D of domain elements, and a finite set of constraints. A constraint defines for its scope $V' \subseteq V$ the allowed instantiations of the variables in V' by values in D. The question is if there exists an instantiation of all variables such that no constraint of the instance is violated.

Although solving CSPs is known to be NP-hard in general, many problems that arise in practice have particular properties that allow them to be solved efficiently. The question of identifying restrictions to the general problem that are sufficient to ensure tractability is important from both a theoretical and a practical point of view. Such restrictions may either involve the *nature* of the constraints (i.e., which instantiations of the variables are allowed) or they may involve the *structure* of the constraints. In this paper we consider the second approach. The structure of a CSP instance can be modeled by its *constraint hypergraph*. Hypergraphs are a generalization of graphs where each edge (called *hyperedge*) connects an arbitrary subset of vertices. Let V be the set of variables and E be the set of constraint scopes of some CSP instance. Then the constraint hypergraph of this instance is given by $H = (V, E)$.

⋆ Corresponding author: Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, England, UK, E-Mail: georg.gottlob@comlab.ox.ac.uk

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 1–11, 2008.

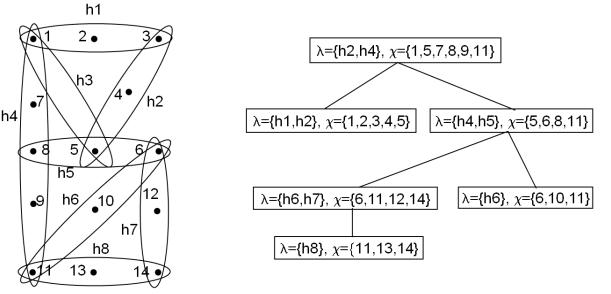


Fig. 1. A hypergraph and its generalized hypertree decomposition

A prominent tractable class of CSPs are *acyclic* CSPs. A CSP instance is acyclic if and only if its constraint hypergraph is acyclic. If a CSP instance is acyclic, then it can be solved efficiently by Yannakakis's classical algorithm [28]. The favorable results on acyclic CSPs extend to classes of "nearly acyclic" CSPs. Several methods have been suggested in the literature to transform an arbitrary CSP instance into an acyclic one, making the vague notion of "nearly acyclic" precise. The most prominent methods include: *tree clustering* [6], *hinge decomposition* [16,17], *cycle cutset* and *cycle hypercutset* [4,12], *hinge-tree clustering* [12], *query decomposition* [2], and *(generalized) hypertree decomposition* [13]. *Generalized hypertree decomposition* [13] is the most general one [11]. Each of these methods has an associated measure for the degree of acyclicity. In the case of generalized hypertree decomposition, this measure is called *generalized hypertree-width*. The smaller the generalized hypertree-width, the "more acyclic" the CSP is; acyclic CSPs have generalized hypertree-width one.

The formal definition of a generalized hypertree decomposition is given in the following: Let $H = (V(H), E(H))$ be a hypergraph. A *tree decomposition* [25] of H is a tree $T = (V(T), E(T))$ together with a mapping $\chi : V(T) \rightarrow 2^{V(H)}$ labeling each node of the tree with a set of vertices of H such that (i) for each $e \in E(H)$ there is a node $t \in V(T)$ with $e \subseteq \chi(t)$ and (ii) for each $v \in V(H)$ the set $\{t \in V(T) \mid v \in \chi(t)\}$ induces a connected subtree. The *width* of a tree decomposition is the maximum $|\chi(t)|-1$ over all nodes $t \in V(T)$. The *treewidth* of a hypergraph is the minimum width over all its tree decompositions. A *generalized hypertree decomposition* [13] of H is a tree decomposition (T, χ) of H together with a mapping $\lambda : V(T) \rightarrow 2^{E(H)}$ labeling each node of the tree with a set of hyperedges of H such that for each $t \in V(T)$ it holds that $\chi(t) \subseteq \bigcup \lambda(t)$. The *width* of a generalized hypertree decomposition is the maximum $|\lambda(t)|$ over all nodes $t \in V(T)$. The *generalized hypertree-width* of a hypergraph is the minimum width over all its generalized hypertree decompositions. We say that a tree decomposition and a generalized hypertree decomposition is *optimal* if it has minimal width. Figure 1 shows a hypergraph H and one of its generalized hypertree decompositions. The width of the underlying tree decomposition is 5, and the width of the generalized hypertree decomposition is 2.

It has been an open question whether optimal generalized hypertree decompositions can be computed in polynomial time if the width is bounded by some constant. This question was recently answered negatively, i.e., computing optimal generalized hypertree decompositions is NP-hard even for bounded width [14]. Thus, in order to enforce

a polynomial runtime in the case of bounded width, an additional condition has been added to the definition of generalized hypertree decompositions: A *hypertree decomposition* [13] of H is a rooted generalized hypertree decomposition of H such that for each $t \in V(T)$ it holds that $(\bigcup \lambda(t)) \cap (\bigcup_{t' \in V(T_t)} \chi(t')) \subseteq \chi(t)$, where T_t denotes the subtree of T rooted at t. Although optimal hypertree decompositions can be computed in polynomial time for bounded width, the degree of the polynomial in the runtime estimation depends on the width. Thus, known algorithms for computing optimal hypertree decompositions are only practical for small bounds on the width [15]. For this reason we consider heuristic approaches for computing (not necessarily optimal) hypertree decompositions of small width. Since these heuristic approaches do not need the additional condition of a hypertree decomposition, we actually construct *generalized* hypertree decompositions that potentially allow smaller widths. Note that a generalized hypertree decomposition suffices to solve the corresponding CSP instance in polynomial time. For simplicity, we will write hypertree decomposition instead of generalized hypertree decomposition in the remainder of this paper.

Heuristically constructed hypertree decompositions are not necessarily optimal. However, the smaller the width of the obtained hypertree decomposition, the faster the corresponding CSP instance can be solved. In fact, a CSP instance can be solved based on its hypertree decomposition as follows: For each node t of the hypertree, all constraints in $\lambda(t)$ are "joined" into a new constraint over the variables in $\chi(t)$. For bounded width, i.e., for bounded cardinality of $\lambda(t)$, this yields a polynomial time reduction to an equivalent acyclic CSP instance which can be solved by Yannakakis's algorithm [28].

Several heuristic approaches for the construction of hypertree decompositions have been proposed in the literature: In [22] an approach based on the vertex connectivity of the given hypergraph (in terms of its primal graph and incidence graph) has been presented, and in [26] the application of branch decomposition heuristics has been considered. Moreover, in [24] the application of genetic algorithms has been investigated.

The problem of these approaches, however, is that they work only well on a restricted class of benchmark examples or they show a very poor runtime performance even on rather small examples. In this paper we present new heuristic approaches that run very fast on a large variety of benchmark examples from both industry and academics; moreover, the results of our algorithms are in most cases superior to those of previous approaches, and even if they are not better than previous results, they are very close to them.

This paper is organized as follows: In Section 2 we present the application of tree decomposition and set cover heuristics, and in Section 3 we present the application of various hypergraph partitioning heuristics. Finally, we evaluate our algorithms experimentally in Section 4, and we conclude in Section 5. Note that due to space limitations we can only succinctly describe our approaches.

2 Tree Decomposition and Set Cover

In this section we combine heuristics for constructing tree decompositions based on linear vertex orderings with set cover heuristics. Recall that the definition of a hypertree

decomposition can be divided into two parts: (i) the definition of a tree decomposition (T, χ) and (ii) the introduction of λ such that $\chi(t) \subseteq \bigcup \lambda(t)$ for every node t. Since the χ-labels contain vertices of the underlying hypergraph and the λ-labels contain hyperedges, i.e., sets of vertices, of the underlying hypergraph, it is clear that the condition in (ii) is nothing else but covering the vertices in $\chi(t)$ by hyperedges in $\lambda(t)$ for every node t. In other words, if we are given a tree decomposition, we can simply construct a hypertree decomposition by putting for each node t hyperedges into $\lambda(t)$ that cover the vertices in $\chi(t)$. This is the basic idea of a heuristic hypertree decomposition approach [23] originally (but somehow misleadingly) named after the CSP solving technique *bucket elimination (BE)* [5]. The heuristic assumption behind this approach is that (heuristically obtained) tree decompositions of small width allow hypertree decompositions of small width by applying set cover heuristics.

The literature provides several powerful tree decomposition heuristics. An important class of such heuristics is based on finding an appropriate linear ordering of the vertices from which a tree decomposition can be constructed [1]. We use the following three well-known vertex ordering heuristics [1,5,23]: *Maximum cardinality*, *minimum induced-width* (also known as *minimum degree*), and *minimum fill-in*. We construct a tree decomposition based on vertex orderings for each of these ordering heuristics and compute the λ-labels by applying the following two set cover heuristics: The first one iteratively picks a hyperedge that covers the largest number of uncovered vertices. The second one assigns the weight 0 to each covered vertex and the weight $1 - \frac{m}{n}$ to each uncovered vertex, where m is the number of hyperedges containing the vertex and n is the number of hyperedges in the hypergraph. The heuristic iteratively picks the hyperedge that has the highest weighted sum over all its vertices. In both heuristics, ties are broken randomly. At each node t we apply both heuristics and define $\lambda(t)$ as the smaller set.

We also considered an approach dual to the above one in the sense that we apply the above steps to the dual hypergraph. The dual hypergraph of a hypergraph is simply obtained by swapping the roles of hyperedges and vertices. For symmetry reasons, let us call this approach *dual bucket elimination (DBE)*. Our intuition for using the dual hypergraph instead of the original hypergraph is that the ordering heuristics aim at minimizing the labeling sets, which are the χ-labels in the case of the original hypergraph. However, the width of a hypertree decomposition is determined by the size of the λ-labels. So our aim was to apply the ordering heuristics in order to minimize the λ-labels, which is exactly what is done when applying BE to the dual hypergraph. Our procedure is the following: (i) build the dual hypergraph, (ii) apply BE to construct a tree decomposition, (iii) interpret the labeling sets as λ-labels of a hypertree, and (iv) set the χ-labels appropriately in a straightforward way (see e.g. [13]). The resulting hypertree is then a hypertree decomposition of the original hypergraph.

3 Hypergraph Partitioning

In this section we consider the use of hypergraph partitioning heuristics that aim at finding a partitioning of the vertices of the hypergraph that is optimal w.r.t. a valuation function on the set of hyperedges connecting the partitions while, at the same time, being subject to restrictions on the relative sizes of the partitions. As hypergraph

partitioning with restrictions on the sizes of the partitions is NP-complete [9], various successful heuristic methods have been proposed in the literature. In the following we consider the applicability of an algorithm due to Fiduccia and Mattheyses [7] as well as the hMETIS library [18], and we propose a new heuristic for hypergraph partitioning based on tabu search. Note that our construction of hypertree decompositions using hypergraph partitioning heuristics is itself heuristic since the computation of cuts needed for such a construction is NP-hard and not even fixed-parameter tractable [27].

Korimort [22] was the first who applied hypergraph partitioning heuristics for hypertree decomposition by recursively separating the hypergraph into smaller and smaller subgraphs. In each partitioning step, a new hypertree node is created and labeled with a set of hyperedges (called *separator*) disconnecting the subgraphs; the χ-labels can be computed in a straightforward way as described in [13]. Note, however, that a hypertree decomposition cannot be constructed by simply connecting the obtained nodes according to the partitioning tree since such a decomposition does not necessarily satisfy the connectedness condition. For this reason Korimort suggested to add a *special hyperedge* to each subgraph containing the vertices in the intersection between the subgraphs in order to enforce their joint appearance in the χ-label of a later generated node. This node can then be used as the root of the subtree. See [22] for more details.

Unfortunately, the introduction of special hyperedges raises the problem of how to evaluate a cut whose separator contains such hyperedges. Being not contained in the original hypergraph, they have to be replaced by possibly more than one hyperedge in the final decomposition which might increase the size of the λ-label of the corresponding node, and thus the width of the decomposition.

To address this problem, we considered hyperedges with associated integral weights and three different weighting schemes; the cut can be uniformly evaluated as the sum of the weights of all hyperedges in the separator. All weighting schemes assign the weight one to all ordinary hyperedges contained in the original hypergraph. The weighting scheme (W1) also assigns the weight one to all special hyperedges, thus treating them like ordinary hyperedges. In the weighting scheme (W+) the weight assigned to a special hyperedge equals the number of ordinary hyperedges needed to cover the vertices of the special hyperedge. Obviously, this might lead to an inaccurate evaluation of separators containing more than a single special hyperedge. As a compromise, we considered the scheme (W2) assigning the weight two to all special hyperedges.

3.1 Partitioning Using Fiduccia-Mattheyses

The hypergraph partitioning algorithm proposed by Fiduccia and Mattheyses [7] is based on an iterative refinement heuristic. First, the hypergraph is arbitrarily partitioned into two parts. In the following, during a sequence of passes, the partitioning is optimized by successively moving vertices to the opposite partition. The decision which vertex is to be moved next is based on a balancing constraint and the *gain* associated to each vertex. The gain is a measure for the impact of the move on the size of the separator. Moreover, a locking mechanism prevents vertices from being moved twice during a pass. Although the algorithm chooses the best possible move in each step, it is nevertheless capable of climbing out of local minima since even the best possible move might lead to a worse solution. A pass is finished after all vertices have been moved once, and

the best solution seen during the pass is taken as the initial solution for the next pass. The algorithm terminates if the initial solution could not be improved during a pass.

The significance of the Fiduccia-Mattheyses algorithm (FM) is due to the possibility of efficient gain updates both at the beginning of a pass and, even more important, during a pass, after a move has been made. Obviously, the gain updates are still efficient if instead of considering the size of the cut we use the sum of the weights of the hyperedges of the cut as a cost function.

Following a note by Korimort [22] stating that considering separators containing special hyperedges should make the problem of finding a good decomposition harder and should not lead to better results, we implemented a variant where all vertices contained in a special hyperedge are moved at once which avoids the valuation problem caused by such hyperedges. However, our results do not support Korimort's assumption.

3.2 Partitioning Using hMETIS

Another approach is based on hMETIS [18,19,20,21], a software package for partitioning hypergraphs developed at the University of Michigan, which is claimed to be one of the best available packages for hypergraph partitioning.

The algorithm follows a multilevel-approach and is organized in three phases. During a first *coarsening phase*, the algorithm constructs a sequence of smaller and smaller hypergraphs by merging vertices and hyperedges. The manual makes no definite statement about the stop criterion, but it mentions that the number of vertices of the coarsest hypergraph is usually between 100 and 200. In the second phase, a variant of the FM algorithm is used to create several bipartitions of this coarsest hypergraph starting from random bipartitions. In the following *uncoarsening phase*, the bipartitions are successively projected onto the next level's finer hypergraph and optimized around the cut. At the end, after having reached the original hypergraph, the best bipartition is chosen.

Unlike the original algorithm of Fiduccia and Mattheyses we implemented, the two routines offered by the hMETIS package are capable of computing k-way partitionings. The routines allow for control over several parameters influencing the phases of the partitioning, however, the interdependencies are non-trivial and the exact impact is hard to estimate. Several tests showed that the parameters with most influence on the width of the resulting hypertree decomposition are the number of partitions (*nparts*), which corresponds to the number of children of the current node in the hypertree decomposition, and the imbalance factor (*ubfactor*) controlling the relative size of the partitions and thus directly influencing the balance of the resulting hypertree. For a complete description of all parameters see [19]. Our test results show that the hypertree decompositions tend to get better for $nparts > 2$, and usually higher *ubfactors* lead to decompositions of smaller width.

3.3 Partitioning Using Tabu Search

In this section we present a new hypergraph partitioning algorithm based on the ideas of tabu search [10]. The algorithm starts with a simple initial solution where one partition contains only one vertex and the second partition contains all other vertices. In each iteration the hyperedges connecting the partitions are considered and the neighborhood

of the current solution is generated by moving vertices from one partition into the other partition. This is done by choosing one of two strategies in each iteration: The first one is chosen with probability p; in this case we move the vertices of a single randomly selected connecting hyperedge. The second one is chosen with probability $1 - p$; in this case we move the vertices of all connecting hyperedges. After the neighborhood is generated, the solutions are evaluated according to a fitness function. The fitness function is the sum of weights over all connecting hyperedges; we tried several weights for these hyperedges. The best solution from the neighborhood, if it is not tabu, becomes the current solution in the next iteration. If the best solution from the neighborhood is tabu, then the *aspiration criterion* is applied. For the aspiration criterion, we use a standard version [10] according to which the tabu status of a move is ignored if the move yields a solution that is better than the currently best solution. For finding the most appropriate length of the tabu list, we tried several lengths and probabilities p. In particular, we tried the following lengths for the tabu list depending on the number $|V|$ of vertices in the hypergraph: $\frac{|V|}{2}$, $\frac{|V|}{3}$, $\frac{|V|}{5}$, $\frac{|V|}{10}$.

4 Experimental Results

In this section we report the experimental results obtained by our implementations of the heuristics described in this paper. We use hypergraphs from the CSP hypergraph library [8] as benchmarks. This collection contains various classes of constraint hypergraphs from industry (DaimlerChrysler, NASA, and ISCAS) as well as synthetically generated ones (Grids and Cliques). The CSP hypergraph library and the executables of the current implementations can be downloaded from the DBAI Hypertree Project website [3]. All experimental results were obtained on a machine with Intel Xeon (dual) processor, 2.2 GHz, 2 GB main memory. Since ties are broken randomly in our algorithms, we applied each method five times to all instances. The results are presented in Table 1. There are two columns for each method: the minimal width obtained during five runs and the average runtime in seconds over five runs. We emphasized the minimal widths in each row by bold numbers, which makes it easy to identify the algorithms that perform best on each class of benchmarks.

The first two columns (BE and DBE) are based on tree decomposition and set cover heuristics. Our experiments show that BE clearly outperforms DBE. Only on the Clique hypergraphs, DBE is faster than BE; this however is not surprising since the number of hyperedges (i.e., the number of vertices of the dual hypergraph) is much smaller than the number of vertices for all instances in this class.

The last four columns (FM-W1, TS-W2, HM-W2, HM-best) are based on hypergraph partitioning heuristics. The results for FM are obtained using weighting scheme (W1), i.e., we do not distinguish between ordinary and special hyperedges. The results for TS and HM are obtained using weighting scheme (W2). We found out that, in most cases, the weighting schemes (W1) and (W2) yield similar results which are significantly better than those achieved when using scheme (W+). The column HM-best displays the minimal width obtained by the HM algorithm over all three weighting schemes together with the runtime for the selected weighting scheme. Our experiments show that HM clearly outperforms the other partitioning based approaches. An

explanation for this is that hMETIS is a highly optimized library that combines several hypergraph partitioning techniques. Seemingly this also influences the results when using hMETIS for computing hypertree decompositions.

Comparing all approaches proposed in this paper, we conclude that the best results are obtained by BE and HM. BE outperforms HM on the classes DaimlerChrysler, NASA, ISCAS, and Cliques, whereas HM outperforms BE on the class Grids. In particular, BE is superior on all industrial benchmark examples. Concerning the runtime, Table 1 shows that all our algorithms are fast. Note, moreover, that the runtime of BE is especially low on almost all benchmark classes.

Compared to previous heuristic hypertree decomposition algorithms [22,26,24], our new methods demonstrate a significant improvement. Two previous approaches [22,26] are clearly outperformed by BE; they have been evaluated on the DaimlerChrysler benchmark class where BE returns the same or better results in less than one second. The third approach [24] builds up on an unpublished preliminary version [23] of our algorithm described in Section 2. In particular, the authors apply genetic algorithms in order to find linear vertex orderings that lead to hypertree decompositions of smaller width. In this way they are able to slightly improve our results for ten of our hypergraph examples; for other examples they get the same or even worse results. However,

Table 1. Experimental results obtained by Bucket Elimination (BE, DBE), Fiduccia-Mattheyses (FM), Tabu Search (TS), and hMETIS (HM)

	BE		DBE		FM (W1)		TS (W2)		HM (W2)		HM (best)	
Instance (Vertices/Edges)	width	time	width	time	width	time	width	time	width	time	width	time
adder_15 (106/76)	**2**	0s	**2**	0s	**2**	0s	4	0s	**2**	3s	**2**	3s
adder_25 (176/126)	**2**	0s	**2**	0s	**2**	1s	4	0s	**2**	7s	**2**	6s
adder_50 (351/251)	**2**	0s	**2**	0s	**2**	6s	4	1s	**2**	13s	**2**	12s
adder_75 (526/376)	**2**	0s	**2**	0s	**2**	21s	5	2s	**2**	21s	**2**	19s
adder_99 (694/496)	**2**	0s	**2**	0s	**2**	53s	5	3s	**2**	28s	**2**	25s
bridge_15 (137/137)	**3**	0s	**3**	0s	8	1s	8	1s	4	7s	**3**	6s
bridge_25 (227/227)	**3**	0s	**3**	0s	13	1s	6	1s	4	11s	**3**	11s
bridge_50 (452/452)	**3**	0s	**3**	0s	29	5s	10	3s	4	24s	4	22s
bridge_75 (677/677)	**3**	0s	**3**	0s	44	10s	10	5s	4	39s	**3**	35s
bridge_99 (893/893)	**3**	0s	**3**	0s	64	18s	10	7s	4	48s	4	45s
NewSystem1 (142/84)	**3**	0s	**3**	0s	4	1s	6	1s	4	5s	**3**	5s
NewSystem2 (345/200)	**4**	0s	**4**	0s	9	2s	6	2s	**4**	14s	**4**	13s
NewSystem3 (474/278)	**5**	0s	**5**	0s	17	4s	11	4s	**5**	19s	**5**	18s
NewSystem4 (718/418)	**5**	0s	**5**	0s	22	8s	12	7s	**5**	31s	**5**	29s
atv_partial_system (125/88)	**3**	0s	4	0s	4	0s	5	1s	4	6s	4	6s
NASA (579/680)	**21**	0s	56	13s	56	20s	98	34s	33	90s	32	84s
c432 (196/160)	**9**	0s	**9**	0s	15	3s	24	4s	13	20s	12	19s
c499 (243/202)	**13**	0s	20	0s	18	3s	27	5s	18	30s	17	28s
c880 (443/383)	**19**	0s	25	0s	31	8s	41	7s	29	50s	25	46s
c1355 (587/546)	**13**	0s	22	0s	32	10s	55	14s	22	66s	22	61s
c1908 (913/880)	32	0s	33	1s	65	23s	70	26s	**29**	86s	**29**	77s
c2670 (1350/1193)	**33**	0s	35	1s	66	56s	78	46s	38	119s	38	106s
c3540 (1719/1669)	**63**	2s	73	11s	97	133s	129	104s	73	166s	73	149s
c5315 (2485/2307)	**44**	3s	61	24s	120	250s	157	157s	72	242s	68	214s
c6288 (2448/2416)	**41**	10s	45	77s	148	478s	329	245s	45	210s	45	186s
c7552 (3718/3512)	37	4s	**35**	8s	161	514s	188	351s	37	365s	37	309s
s27 (17/13)	**2**	0s	**2**	0s	**2**	0s	3	0s	**2**	0s	**2**	0s
s208 (115/104)	**7**	0s	**7**	0s	**7**	1s	11	1s	**7**	10s	**7**	9s
s298 (139/133)	**5**	0s	8	0s	7	1s	17	2s	7	11s	6	10s
s344 (184/175)	**7**	0s	8	0s	8	2s	12	2s	8	21s	**7**	19s

Table 1. (*continued*)

Instance (Vertices/Edges)	BE		DBE		FM (W1)		TS (W2)		HM (W2)		HM (best)	
	width	time	width	time	width	time	width	time	width	time	width	time
s349 (185/176)	**7**	0s	9	0s	8	1s	12	2s	9	21s	**7**	19s
s382 (182/179)	**5**	0s	8	0s	7	2s	17	2s	8	16s	7	15s
s386 (172/165)	**8**	0s	15	0s	13	2s	26	3s	11	16s	11	15s
s400 (186/183)	**6**	0s	8	0s	8	2s	18	2s	8	18s	7	17s
s420 (231/212)	**9**	0s	**9**	0s	10	2s	14	2s	10	29s	10	24s
s444 (205/202)	**6**	0s	8	0s	8	2s	25	3s	8	21s	8	20s
s510 (236/217)	**23**	0s	31	0s	**23**	4s	41	4s	27	27s	27	25s
s526 (217/214)	**8**	0s	13	0s	13	2s	32	3s	11	29s	11	27s
s641 (433/398)	**7**	0s	14	0s	19	5s	21	5s	14	31s	14	28s
s713 (447/412)	**7**	0s	13	0s	21	5s	25	5s	14	33s	14	31s
s820 (312/294)	**12**	0s	27	2s	23	8s	77	9s	24	42s	19	38s
s832 (310/292)	**12**	0s	28	2s	22	8s	71	10s	26	42s	20	39s
s838 (457/422)	**15**	0s	**15**	0s	19	6s	24	6s	**15**	55s	**15**	46s
s953 (440/424)	**39**	0s	53	1s	50	18s	70	12s	45	52s	45	47s
s1196 (561/547)	**34**	0s	53	2s	50	22s	73	15s	43	69s	43	62s
s1238 (540/526)	**34**	0s	56	3s	56	20s	80	18s	43	66s	43	59s
s1423 (748/731)	**18**	0s	22	1s	29	25s	54	19s	27	78s	26	71s
s1488 (667/659)	**23**	0s	77	26s	45	46s	148	36s	39	85s	39	77s
s1494 (661/653)	**23**	0s	78	27s	49	45s	150	38s	38	85s	36	77s
s5378 (2993/2958)	**87**	7s	108	23s	178	308s	169	271s	89	279s	89	246s
b01 (47/45)	**5**	0s	6	0s	**5**	0s	10	0s	**5**	2s	**5**	2s
b02 (27/26)	**3**	0s	5	0s	4	0s	7	0s	4	1s	4	1s
b03 (156/152)	**7**	0s	11	0s	11	1s	16	2s	9	15s	8	14s
b04 (729/718)	**26**	0s	39	2s	44	26s	69	26s	38	82s	35	73s
b05 (962/961)	**18**	0s	29	1s	42	33s	70	30s	32	114s	32	99s
b06 (50/48)	**5**	0s	6	0s	**5**	0s	12	0s	**5**	3s	**5**	2s
b07 (433/432)	**19**	0s	29	0s	38	7s	59	10s	33	57s	31	54s
b08 (179/170)	**10**	0s	13	0s	14	2s	20	2s	12	21s	12	19s
b09 (169/168)	**10**	0s	12	0s	13	2s	20	2s	12	20s	12	19s
b10 (200/189)	**14**	0s	18	0s	17	3s	33	3s	16	24s	16	21s
b11 (764/757)	**31**	0s	47	3s	65	28s	98	27s	38	89s	38	79s
b12 (1070/1065)	**27**	0s	39	3s	38	55s	83	39s	34	111s	34	102s
b13 (352/342)	9	0s	10	0s	10	4s	17	4s	**8**	36s	**8**	33s
grid2d_10 (50/50)	**5**	0s	6	0s	**5**	0s	8	0s	**5**	3s	**5**	3s
grid2d_15 (113/112)	**9**	0s	**9**	0s	10	1s	12	1s	10	11s	10	10s
grid2d_20 (200/200)	12	0s	**11**	0s	15	2s	18	2s	14	29s	12	28s
grid2d_25 (313/312)	**15**	0s	**15**	0s	18	5s	26	5s	**15**	50s	**15**	43s
grid2d_30 (450/450)	19	0s	20	0s	21	11s	29	8s	**16**	70s	**16**	58s
grid2d_35 (613/612)	23	0s	23	0s	30	20s	41	13s	**19**	87s	**19**	73s
grid2d_40 (800/800)	26	0s	25	0s	28	38s	41	20s	**22**	108s	**22**	91s
grid2d_45 (1013/1012)	31	1s	30	1s	40	58s	47	31s	**25**	130s	**25**	109s
grid2d_50 (1250/1250)	33	1s	32	1s	44	88s	52	41s	**28**	154s	**28**	130s
grid2d_60 (1800/1800)	42	2s	39	3s	55	203s	75	75s	**34**	209s	**34**	178s
grid2d_70 (2450/2450)	49	4s	47	4s	65	347s	65	119s	**41**	283s	**41**	239s
grid2d_75 (2813/2812)	52	6s	50	7s	70	504s	99	158s	**44**	324s	**44**	274s
grid3d_4 (32/32)	**6**	0s	**6**	0s	**6**	0s	12	0s	**6**	1s	**6**	1s
grid3d_5 (63/62)	9	0s	10	0s	**8**	1s	18	1s	11	4s	10	3s
grid3d_6 (108/108)	14	0s	14	0s	**12**	1s	25	2s	15	9s	14	9s
grid3d_7 (172/171)	20	0s	20	0s	18	2s	33	5s	19	27s	**16**	24s
grid3d_8 (256/256)	25	0s	27	0s	25	5s	44	9s	21	48s	**20**	40s
grid3d_9 (365/364)	34	0s	26	0s	34	9s	56	14s	**24**	67s	**24**	56s
grid3d_10 (500/500)	42	1s	40	1s	41	20s	67	26s	**31**	93s	**31**	77s
grid3d_11 (666/665)	52	1s	53	2s	40	36s	83	43s	**37**	119s	**37**	99s
grid3d_12 (864/864)	63	3s	62	3s	53	61s	98	66s	45	150s	**44**	127s
grid3d_13 (1099/1098)	78	5s	68	6s	60	107s	122	101s	**53**	186s	**53**	158s
grid3d_14 (1372/1372)	88	10s	93	10s	86	161s	176	162s	**69**	230s	**69**	196s
grid3d_15 (1688/1687)	104	15s	103	15s	93	253s	151	245s	**76**	278s	**76**	244s
grid3d_16 (2048/2048)	120	24s	131	24s	100	400s	174	328s	87	339s	**82**	303s
grid4d_3 (41/40)	**8**	0s	**8**	0s	**8**	0s	20	0s	9	2s	**8**	2s
grid4d_4 (128/128)	**17**	0s	18	0s	**17**	1s	40	3s	19	13s	18	12s
grid4d_5 (313/312)	35	0s	37	0s	32	8s	78	17s	**28**	58s	**28**	48s
grid4d_6 (648/648)	64	3s	71	2s	58	40s	140	67s	**47**	123s	**47**	106s
grid4d_7 (1201/1200)	109	14s	110	14s	89	134s	182	194s	74	229s	**71**	208s
grid4d_8 (2048/2048)	164	62s	166	62s	120	441s	310	581s	**107**	408s	**107**	393s
grid5d_3 (122/121)	**18**	0s	20	0s	**18**	1s	49	4s	20	11s	19	10s

Table 1. (*continued*)

Instance (Vertices/Edges)	BE		DBE		FM (W1)		TS (W2)		HM (W2)		HM (best)	
	width	time	width	time	width	time	width	time	width	time	width	time
grid5d_4 (512/512)	63	2s	68	2s	49	25s	137	56s	**46**	92s	**46**	78s
grid5d_5 (1563/1562)	161	42s	159	42s	118	280s	362	474s	**111**	328s	**111**	319s
clique_10 (45/10)	**5**	0s	**5**	0s	**5**	0s	6	0s	**5**	0s	**5**	0s
clique_15 (105/15)	**8**	0s	**8**	0s	12	0s	**8**	1s	**8**	1s	**8**	1s
clique_20 (190/20)	**10**	0s	**10**	0s	20	0s	11	3s	**10**	1s	**10**	1s
clique_25 (300/25)	**13**	0s	**13**	0s	25	0s	14	8s	**13**	2s	**13**	2s
clique_30 (435/30)	**15**	1s	**15**	0s	30	0s	16	16s	**15**	3s	**15**	3s
clique_35 (595/35)	**18**	2s	**18**	0s	35	0s	19	29s	**18**	5s	**18**	5s
clique_40 (780/40)	**20**	5s	**20**	0s	40	0s	22	51s	**20**	6s	**20**	6s
clique_45 (990/45)	**23**	10s	**23**	0s	45	1s	24	80s	**23**	10s	**23**	9s
clique_50 (1225/50)	**25**	19s	**25**	0s	50	1s	28	145s	**25**	15s	**25**	13s
clique_60 (1770/60)	**30**	60s	**30**	0s	60	2s	34	340s	59	1s	50	1s
clique_70 (2415/70)	**35**	155s	**35**	0s	70	4s	39	601s	68	2s	67	1s
clique_75 (2775/75)	**38**	239s	**38**	0s	75	6s	41	920s	71	3s	71	2s
clique_80 (3160/80)	**40**	350s	**40**	0s	80	8s	43	1248s	76	4s	72	3s
clique_90 (4005/90)	**45**	724s	**45**	2s	90	12s	50	2045s	89	8s	78	5s
clique_99 (4851/99)	**50**	1280s	**50**	4s	99	19s	54	2845s	99	14s	97	8s

the runtime in their approach increases drastically. For instance, the authors obtained a hypertree decomposition of width 19 (-2) for the NASA example but the runtime to obtain this result was more than 17 hours.

5 Conclusion

In this paper we presented two generic heuristic approaches for the construction of generalized hypertree decompositions. The first one is based on tree decomposition and set cover heuristics, and the second one is based on hypergraph partitioning heuristics. We evaluated our algorithms empirically on a variety of benchmark examples and could show that our approaches clearly improve previous approaches for this problem. Future research is to enhance our techniques in order to obtain hypertree decompositions of smaller width but without increasing the runtime of the algorithms excessively.

Acknowledgments. This research was supported by the Austrian Science Fund (FWF), project P17222-N04. The work of Tobias Ganzow and Ben McMahan has been carried out while visiting the Technische Universität Wien. The visit of Tobias Ganzow was funded by the Wolfgang Pauli Institute (WPI).

References

1. Bodlaender, H.L.: Discovering treewidth. In: Vojtáš, P., Bieliková, M., Charron-Bost, B., Sýkora, O. (eds.) SOFSEM 2005. LNCS, vol. 3381, pp. 1–16. Springer, Heidelberg (2005)
2. Chekuri, C., Rajaraman, A.: Conjunctive query containment revisited. Theoretical Computer Science 239(2), 211–229 (2000)
3. DBAI Hypertree Project website, Vienna University of Technology, http://www.dbai.tuwien.ac.at/proj/hypertree/
4. Dechter, R.: Constraint networks. In: Encyclopedia of Artificial Intelligence, 2nd edn., vol. 1, pp. 276–285. Wiley and Sons, Chichester (1992)

5. Dechter, R.: Constraint Processing. Morgan Kaufmann, San Francisco (2003)
6. Dechter, R., Pearl, J.: Tree clustering for constraint networks. Artificial Intelligence 38(3), 353–366 (1989)
7. Fiduccia, C.M., Mattheyses, R.M.: A linear-time heuristic for improving network partitions. In: Proc. of the 19th Conference on Design Automation (DAC 1982), pp. 175–181. IEEE Press, Los Alamitos (1982)
8. Ganzow, T., Gottlob, G., Musliu, N., Samer, M.: A CSP hypergraph library. Technical Report, DBAI-TR-2005-50, Vienna University of Technology (2005)
9. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman and Co., New York (1979)
10. Glover, F., Laguna, M.: Tabu search. Kluwer Academic Publishers, Dordrecht (1997)
11. Gottlob, G., Leone, N., Scarcello, F.: A comparison of structural CSP decomposition methods. Artificial Intelligence 124(2), 243–282 (2000)
12. Gottlob, G., Leone, N., Scarcello, F.: The complexity of acyclic conjunctive queries. Journal of the ACM 48(3), 431–498 (2001)
13. Gottlob, G., Leone, N., Scarcello, F.: Hypertree decomposition and tractable queries. Journal of Computer and System Sciences 64(3), 579–627 (2002)
14. Gottlob, G., Miklós, Z., Schwentick, T.: Generalized hypertree decompositions: NP-hardness and tractable variants. In: Proc. of the 26th ACM Symposium on Principles of Database Systems (PODS 2007), pp. 13–22. ACM Press, New York (2007)
15. Gottlob, G., Samer, M.: A backtracking-based algorithm for hypertree decomposition. ACM Journal of Experimental Algorithmics (to appear)
16. Gyssens, M., Jeavons, P.G., Cohen, D.A.: Decomposing constraint satisfaction problems using database techniques. Artificial Intelligence 66(1), 57–89 (1994)
17. Gyssens, M., Paredaens, J.: A decomposition methodology for cyclic databases. In: Gallaire, H., Nicolas, J.-M., Minker, J. (eds.) Advances in Data Base Theory, vol. 2, pp. 85–122. Plemum Press, New York (1984)
18. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: Applications in VLSI domain. IEEE Transactions on Very Large Scale Integration Systems 7(1), 69–79 (1999)
19. Karypis, G., Kumar, V.: hMetis: A hypergraph partitioning package, version 1.5.3 (1998)
20. Karypis, G., Kumar, V.: Multilevel k-way partitioning scheme for irregular graphs. Journal of Parallel and Distributed Computing 48(1), 96–129 (1998)
21. Karypis, G., Kumar, V.: Multilevel k-way hypergraph partitioning. In: Proc. of the 36th ACM/IEEE Conference on Design Automation (DAC 1999), pp. 343–348. ACM Press, New York (1999)
22. Korimort, T.: Heuristic Hypertree Decomposition. PhD thesis, Vienna University of Technology (2003)
23. McMahan, B.: Bucket eliminiation and hypertree decompositions. Implementation Report, Database and AI Group, Vienna University of Technology (2004)
24. Musliu, N., Schafhauser, W.: Genetic algorithms for generalized hypertree decompositions. European Journal of Industrial Engineering 1(3), 317–340 (2007)
25. Robertson, N., Seymour, P.D.: Graph minors. II. Algorithmic aspects of tree-width. Journal of Algorithms 7, 309–322 (1986)
26. Samer, M.: Hypertree-decomposition via branch-decomposition. In: Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), pp. 1535–1536. Professional Book Center (2005)
27. Samer, M., Szeider, S.: Complexity and applications of edge-induced vertex-cuts. Technical Report, arXiv:cs.DM/0607109 (2006)
28. Yannakakis, M.: Algorithms for acyclic database schemes. In: Proc. of the 7th International Conference on Very Large Data Bases (VLDB 1981), pp. 81–94. IEEE Press, Los Alamitos (1981)

Nonlinear Bayesian Filters for Training Recurrent Neural Networks

Ienkaran Arasaratnam and Simon Haykin

Cognitive Systems Laboratory
Department of Electrical & Computer Engineering
McMaster University
Hamilton, ON, L8S 4K1
aienkaran@grads.ece.mcmaster.ca, haykin@mcmaster.ca

Abstract. In this paper, we present nonlinear Bayesian filters for training recurrent neural networks with a special emphasis on a novel, more accurate, derivative-free member of the approximate Bayesian filter family called the cubature Kalman filter. We discuss the theory of Bayesian filters, which is rooted in the state-space modeling of the dynamic system in question and the linear estimation principle. For improved numerical stability and optimal performance during training period, a number of techniques of how to tune Bayesian filters is suggested. We compare the predictability of various Bayesian filter-trained recurrent neural networks using a chaotic time-series. From the empirical results, we conclude that the performance may be greatly improved by the new square-root cubature Kalman filter.

1 Introduction

Neural networks are an important tool in the modern engineer's kit of problem-solvers. They essentially use flexibly parameterized classes of nonlinear functions to approximate input-desired output relations. The parameters, hereafter called *weights*, are unknown to be determined. In the supervised training mode, a training data set is provided with a set of examples, each of which has a distinct input/desired-output. Hence, the objective of the training algorithm is to find these weights so as to match the training data set as closely as possible.

Specifically, recurrent neural networks (RNNs) have had successes in areas such as dynamic system identification [13], nonlinear prediction [7] and control [22]. They are capable of storing temporal dependencies spanning a number of time steps between inputs and desired outputs. To train RNNs, we use the gradient descent method. Here, the gradient descent method is applied in a truncated backpropagation through time setup to compute derivatives of a cost function with respect to weights meaningfully [17], [27]. In the truncated back-propagation through time setup, internal states of a RNN at current time are computed by unfolding recurrent loops to some steps backward in time and propagating states corresponding to that time through the unfolded network. The gradient descent method for training RNNs suffers from the following limitations:

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 12–33, 2008.

(i) Poor convergence (ii) Recency effect (tendency for recent weight update to cause RNNs to forget what they have learned in the past) (iii) Vanishing gradient effect (inability of the gradient method to propagate errors backward deeply). The latter is severe when we use a highly time-correlated signal as the training sequence (e.g., speech signal) [3].

To accelerate the rate of convergence, we may use second-order methods using the Hessian information. Unfortunately, they do not always improve the performance significantly due to Hessians that are intrinsically ill-conditioned. Moreover, their converge rate relies on an accurate evaluation of successive descent directions, which seems meaningful only in a batch of examples. However, updating of weights on an example-by-example (online) basis is preferred when the training data set (i) exhibits non-stationary behavior or (ii) is difficult/expensive to obtain before the training starts. This is where we seek the so-called Bayesian filters.

Bayesian filters utilize Bayes' rule to update weight statistics. To be specific, Bayesian filters track the evolving conditional probability density of weights, given a history of examples. In the recent past, there has been a surge of interest in utilizing nonlinear Bayesian filters for training RNNs [9], [19]. Specifically, they provide the following benefits:

- Bayesian filter-based training algorithms converge more rapidly than the gradient descent method.
- They are well-suited to handle noisy and nonstationary training data.
- The second-order information encoded in the filter-estimated error covariance can often be used to prune the network structure [25].
- Last but by no means least, we mention the benefit of using them in neurocontroller training. The Physics-based models of plants including various non-differential submodels due to dead zones, look-up tables etc., are already in place in an industry. In this case, we may directly utilize derivative-free Bayesian filters for neurocontroller training rather than replacing them with less accurate neural networks using the principle of system identification [19], [26].

Although Bayesian filters achieve better results in fewer iterations than the gradient descent method, this does not necessarily translate to less training time. Of course, their complexity is in the order of w^3 instead of w, where w is the number of weights. As a practical cure, algorithmic implementations included various clever groupings of weights at the expense of 'slightly' degraded performance nearly two decades ago [20], [24]. Fortunately, most of present digital computers are capable of handling this computational demand without seeking any approximative approaches, thanks to the ever-increasing power of computers.

In this paper, one of our objectives is to pique the readers' interest in the Bayesian filter family, namely, the extended Kalman filter (EKF) [1], the central-difference Kalman filter (CDKF) [14] and the *cubature Kalman filter* (CKF) [2], for training networks. However, contributions of this paper mainly lie in how to utilize the CKF, which is novel, more accurate and derivative-free. For example,

- We analyze various Bayesian filters from the optimization perspective. Therein, we go on to prove why the CKF is superior to other filters theoretically and verify it empirically.
- We illustrate how the CKF addresses the following problems: (i) Accommodating a wide range of cost functions, which may be non-differentiable (ii) Vanishing gradient in RNNs: The use of the CKF eliminates the need for unfolding the internal feedback loops of the RNNs in time. The reason is that the CKF is rooted in the philosophy:

  Optimization through integration as opposed to differentiation.
- We discuss how to optimize performance of the CKF by tailoring its various hyperparameters.

The rest of the paper is organized as follows: Section 2 introduces a pair of fundamental equations of the optimal nonlinear Bayesian filters for nonlinear filtering problems. We present the theory of the CKF under the Gaussian assumption in Section 3. Section 4 illustrates that the use of the Bayesian filter-based training is equivalent to a second-order optimization method minimizing the regularized sum of squared error cost function with varying degree of accuracies. Therein, we prove theoretically that the CKF estimate is superior to various other Bayesian filters. In Section 5, we suggest a number of possible extensions and refinements of Bayesian filters with a special emphasis on the CKF for an improved and stable performance in training. We present a computer experiment in Section 6, where we compare the predictability of various Bayesian filter-trained RNNs using a chaotic time-series. Finally, in Section 7, we summarize the paper with some insightful remarks.

2 Optimal Bayesian Solution to Filtering Problems

In this section, we present a theoretically relevant optimal Bayesian solution to nonlinear filtering problems and describe its intractability leading to proliferation of many suboptimal solutions. Consider the problem of sequentially estimating a state vector $\mathbf{x}_k \in \mathbb{R}^{n_x}$ that evolves in discrete-time according to the first-order Markov process:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{q}_{k-1} \tag{1}$$

given a mapping function from the hidden state space to the observable space:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{r}_k, \tag{2}$$

where $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_x}$ and $\mathbf{h} : \mathbb{R}^{n_x} \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ are known; $\mathbf{u}_k \in \mathbb{R}^m$ is the known input at time k; $\mathbf{z}_k \in \mathbb{R}^n$ is the measurement; $\{\mathbf{q}_{k-1}\}$ and $\{\mathbf{r}_k\}$ are uncorrelated process and measurement Gaussian noise sequences with zero means and covariances Q_{k-1} and R_k, respectively. The pair of equations, namely the *process (state transition) equation* (1) and the *measurement equation* (2) are collectively called the state-space model of the dynamic system.

The Bayesian filter extracts information about the hidden state from noisy measurements. Of course, it computes the posterior density of the state, which provides a complete statistical description of the state at that time, in two basic steps:

– *Prediction,* which involves propagating the old posterior density state one time-step ahead before receiving a new measurement. At the end of this step, we obtain the predictive density using Kolmogorov's forward equation:

$$p(\mathbf{x}_k|D_{k-1}) = \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_{k-1}|D_{k-1})p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})d\mathbf{x}_{k-1}, \tag{3}$$

where $D_{k-1} = \{(\mathbf{u}_i, \mathbf{z}_i), i = 1, 2 \ldots (k-1)\}$ denotes the history of input-measurement pairs up to time $(k-1)$; $p(\mathbf{x}_{k-1}|D_{k-1})$ is the old posterior density at time $(k-1)$ and the transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ is obtained from the process equation (1).

– *Correction,* which involves updating the predictive density using the new measurement, $\mathbf{z}_k$. Using Bayes' rule, we write the posterior density:

$$p(\mathbf{x}_k|D_k) = \frac{1}{c_k} p(\mathbf{x}_k|D_{k-1})p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k), \tag{4}$$

where the normalizing constant

$$c_k = \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_k|D_{k-1})p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k)d\mathbf{x}_k, \tag{5}$$

and the measurement likelihood function $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k)$ is obtained from the measurement equation (2).

From the posterior density (4), any statistic regarding $\mathbf{x}_k$ can be computed. For example, we can provide an optimal estimate and its associated covariance (variance) according to 'some' chosen criterion. The optimal Bayesian filter solution given by (3), (4) and (5) provides a unified recursive approach for nonlinear filtering problems conceptually. However, it is not of direct practical importance for two reasons: (i) For a multidimensional system, we require to compute multi-dimensional integrals involved in (3) and (5) (ii) Even after integrals are computed, it may be difficult to propagate relevant statistics through subsequent time steps.

Consequently, an intense research in the past has resulted in a number of suboptimal solutions to nonlinear filtering problems. *Specifically, in machine learning, suboptimal solutions are considered an advantage because they are unlikely to yield an overfitted solution.* In computational terms, the suboptimal solutions to the posterior density can be obtained using one of two approaches:

1. *Local approach.* Here, we derive nonlinear filters by fixing the posterior density to take *a priori* form. The emphasis on locality allows us to work with a limited set of parameters such as central moments of the probability density rather than the probability density itself. For example, we may assume

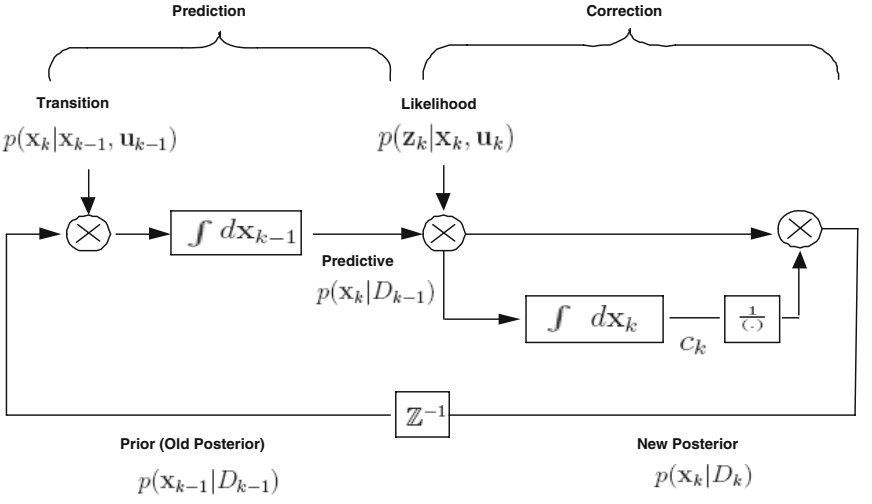


Fig. 1. Signal-flow diagram of the optimal recursive Bayesian filter. ($\mathbb{Z}^{-1}$ denotes the unit time delay operator).

it to be Gaussian, which is completely characterized by mean and variance. Nonlinear filters, namely, the extended Kalman filter (EKF) [1], the central-difference Kalman filter (CDKF) [14], the unscented Kalman filter (UKF) [10], and the cubature Kalman filter (CKF) [2], fall under this first category.

2. *Global approach.* As opposed to the local approach, we do not make any explicit assumption about the posterior density form here. For example, particle filters using Monte Carlo integrations with the importance sampling [4], [5], fall under this second category .

Though the global approaches seem to be more accurate, their accuracy is achieved at the expense of intensive computations. For example, when using particle filters in high-dimensional filtering problems such as training neural networks, a vast majority of particles may be sampled from a non-informative volume of the state-space [12]. Therefore, when the problem dimension increases we use an exponentially increasing number of particles to retain the same accuracy. This is often called the *curse of dimensionality*. In this paper, we focus on supervised training of neural networks, which typically involves a huge number weights to be determined. In this case, we have to be content with a local approach, which makes the design of the filter simple and fast to execute. Next section introduces the CKF using the local approach.

3 Theory of Cubature Kalman Filter

The key assumption of the cubature Kalman filter (CKF) is that the predictive density $p(\mathbf{x}_k|D_{k-1})$ and the filter likelihood density $p(\mathbf{z}_k|D_k)$ are both Gaussian,

which eventually leads to a Gaussian posterior density $p(\mathbf{x}_k|D_k)$. Under this assumption, the cubature Kalman filter solution reduces to how to compute their means and covariances more accurately.

3.1 Prediction

In the prediction step, the CKF computes the mean $\hat{\mathbf{x}}_{k|k-1}$ and the associated covariance $P_{k|k-1}$ of the Gaussian predictive density numerically using cubature rules. We write the predicted mean

$$\hat{\mathbf{x}}_{k|k-1} = \mathbb{E}(\mathbf{x}_k|D_{k-1}), \tag{6}$$

where $\mathbb{E}$ is the statistical expectation operator. Substituting (1) into (6) yields

$$\hat{\mathbf{x}}_{k|k-1} = \mathbb{E}\big[\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{q}_k|D_{k-1}\big]. \tag{7}$$

Because $\mathbf{q}_k$ is assumed to be zero-mean and uncorrelated with the measurement sequence, we get

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbb{E}\big[\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})|D_{k-1}\big] \\ &= \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1}|D_{k-1}) d\mathbf{x}_{k-1} \\ &= \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}) d\mathbf{x}_{k-1},\end{aligned} \tag{8}$$

where $\mathcal{N}(.,.)$ is the conventional symbol for a Gaussian density. Similarly, we obtain the associated error covariance

$$\begin{aligned}P_{k|k-1} &= \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T|\mathbf{z}_{1:k-1}] \\ &= \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \mathbf{f}^T(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}) d\mathbf{x}_{k-1} \\ &\quad - \hat{\mathbf{x}}_{k|k-1} \hat{\mathbf{x}}_{k|k-1}^T + Q_{k-1}.\end{aligned} \tag{9}$$

3.2 Correction

It is known that the innovation process is not only white but also zero-mean Gaussian when the additive measurement noise is Gaussian and the predicted measurement is estimated in the least squares error sense. In this case, we write the predicted measurement density also called the filter likelihood density

$$p(\mathbf{z}_k|D_{k-1}) = \mathcal{N}(\mathbf{z}_k; \hat{\mathbf{z}}_{k|k-1}, P_{zz,k|k-1}), \tag{10}$$

where the predicted measurement and the associated covariance are given by

$$\hat{\mathbf{z}}_{k|k-1} = \int_{\mathbb{R}^{n_x}} \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}) d\mathbf{x}_k \tag{11}$$

$$\begin{aligned}P_{zz,k|k-1} &= \int_{\mathbb{R}^{n_x}} \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \mathbf{h}^T(\mathbf{x}_k, \mathbf{u}_k) \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}) d\mathbf{x}_k \\ &\quad - \hat{\mathbf{z}}_{k|k-1} \hat{\mathbf{z}}_{k|k-1}^T + R_k.\end{aligned} \tag{12}$$

Hence, we write the Gaussian conditional density of the joint state and the measurement:

$$p\left([\mathbf{x}_k^T \;\; \mathbf{z}_k^T]^T | D_{k-1}\right) = \mathcal{N}\left(\begin{pmatrix}\hat{\mathbf{x}}_{k|k-1}\\ \hat{\mathbf{z}}_{k|k-1}\end{pmatrix}, \begin{pmatrix}P_{k|k-1} & P_{xz,k|k-1}\\ P_{xz,k|k-1}^T & P_{zz,k|k-1}\end{pmatrix}\right), \quad (13)$$

where the cross-covariance

$$P_{xz,k|k-1} = \int_{\mathbb{R}^{n_x}} \mathbf{x}_k \mathbf{h}^T(\mathbf{x}_k, \mathbf{u}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1})d\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^T. \quad (14)$$

On the receipt of a new measurement $\mathbf{z}_k$, the CKF computes the posterior density $p(\mathbf{x}_k|D_k)$ from (13) yielding

$$p(\mathbf{x}_k|D_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, P_{k|k}), \quad (15)$$

where

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + G_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \quad (16)$$

$$P_{k|k} = P_{k|k-1} - G_k P_{zz,k|k-1} G_k^T, \quad (17)$$

with the Kalman gain

$$G_k = P_{xz,k|k-1} P_{zz,k|k-1}^{-1}. \quad (18)$$

The signal-flow diagram in Fig. 2 summarizes the steps involved in the recursion cycle of the CKF. The CKF theory reduces to the Kalman filter for a linear Gaussian system case. The CKF numerically computes Gaussian weighted integrals that are present in (8)-(9), (11)-(12) and (14) using cubature rules as outlined next.

3.3 Cubature Rules

In general, cubature rules are constructed to numerically compute multi-dimensional weighted integrals. The CKF specifically uses a third-degree cubature rule to numerically compute Gaussian weighted integrals. The third-degree cubature rule is exact for integrands being polynomials of degree up to three or any odd integer. For example, we use the cubature rule to approximate an n-dimensional Gaussian weighted integral as follows:

$$\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\mathcal{N}(\mathbf{x}; \mu, \Sigma)d\mathbf{x} \approx \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{f}(\mu + \sqrt{\Sigma}\xi_i).$$

where a square-root factor of the covariance Σ satisfies the relationship $\Sigma = \sqrt{\Sigma}\sqrt{\Sigma}^T$ and the set of $2n$ cubature points are given by

$$\xi_i = \begin{cases} \sqrt{n}\mathbf{e}_i, & i = 1, 2 \ldots n \\ -\sqrt{n}\mathbf{e}_{i-n}, & i = n+1, n+2 \ldots 2n. \end{cases}$$

with $\mathbf{e}_i \in \mathbb{R}^n$ denoting the i-th elementary column vector. For a detailed exposition of how to derive cubature points, the reader may consult [2].

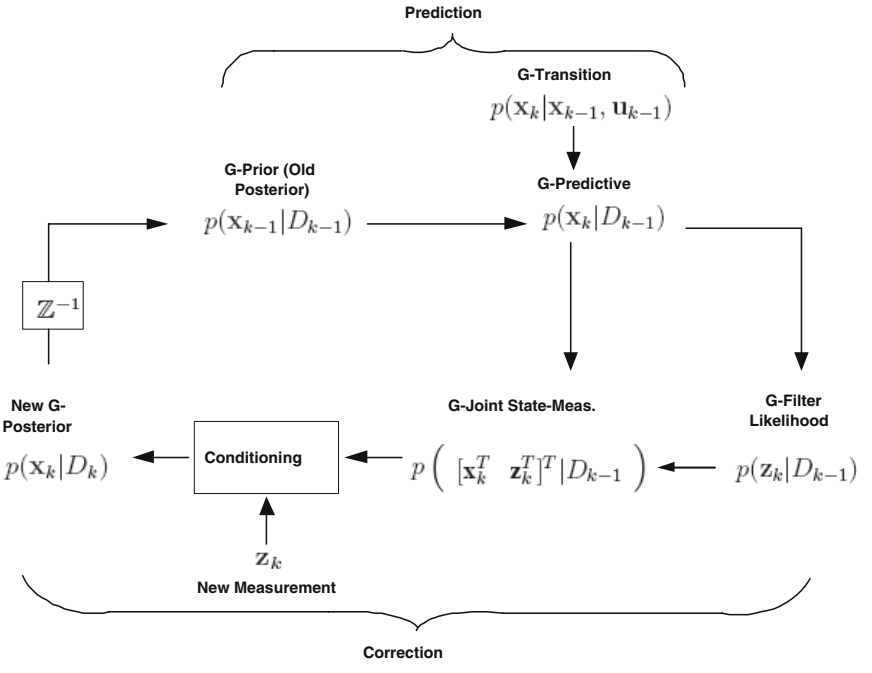


Fig. 2. Signal-flow diagram of the CKF, where 'G-' stands for 'Gaussian-'

4 From Learning to Inference: Bayesian Filters for Supervised Training

Mathematically, we may express the functional form of a neural network as

$$\mathbf{d}_k = \mathbf{h}(\mathbf{w}, \mathbf{u}_k), \tag{19}$$

where $\mathbf{w} \in \mathbb{R}^w$ is the weight vector; $\mathbf{u}_k \in \mathbb{R}^m$ is the input vector; $\mathbf{d}_k \in \mathbb{R}^n$ is the desired output; and the form of the nonlinear function $\mathbf{h}$ is determined by the neural network architecture and its activation functions. We shall assume $\mathbf{u}_k$ to encompass the state variables additionally present in a RNN architecture.

When a set of k examples denoted by $D_k = \{(\mathbf{u}_i, \mathbf{d}_i), i = 1, 2 \ldots k\}$ is given, the objective of the supervised training can be posed as follows:

How do we extract the information about $\mathbf{w}$*, which is contained in* D_k *in 'some' optimal fashion? In the Bayesian estimation paradigm, this readily translates into the question of how to compute the posterior density* $p(\mathbf{w}|D_k)$*?*

To achieve the above objective in the Bayesian paradigm, the first key step is to treat weights to be random variables thereby accounting for uncertainty in their estimate. We assume weight variables to follow the Gaussian random walk model. Hence, we write the state-space model for training neural networks in supervised mode as follows:

$$\text{Process equation:} \quad \mathbf{w}_k = \mathbf{w}_{k-1} + \mathbf{q}_{k-1} \tag{20}$$

$$\text{Measurement equation:} \quad \mathbf{d}_k = \mathbf{h}(\mathbf{w}_k, \mathbf{u}_k) + \mathbf{r}_k, \tag{21}$$

where $\mathbf{w}_k$ may also be interpretable as a *slowly varying parameter* due to the stochastic component $\mathbf{q}_k$, which is assumed to be zero-mean Gaussian process noise with covariance Q_{k-1}; the process noise is purposely injected with decreasing covariance; $\mathbf{r}_k$ is assumed to be zero-mean Gaussian measurement noise with covariance R_k; and the desired output $\mathbf{d}_k$ acts as measurements. Suppose the prior weight statistics are set properly. In this case, the Bayesian filters infer the evolving posterior density of $\mathbf{w}_k$ using the state-space model (20)-(21). The Bayesian filters improve the accuracy of the weight estimate sequentially because they operate on an example-by-example basis.

4.1 Bayesian Filters as Second-Order Optimizers

In this subsection, the nonlinear Bayesian filters under the Gaussian assumption, exemplified by the EKF, the CDKF and the CKF, are viewed to act on the basis of a unified principle: Despite of their algorithmic differences, they all are second-order optimizers in disguise, which minimize a *regularized* sum of squared error cost function with varying degree of accuracies. [1]

Suppose we are given D_k. Under the Gaussian assumption, we write the posterior density (compare (4)):

$$p(\mathbf{w}_k|D_k) = \frac{1}{c_k}\mathcal{N}(\mathbf{w}_k; \hat{\mathbf{w}}_{k|k-1}, P_{k|k-1})\mathcal{N}(\mathbf{t}_k; \mathbf{h}(.,.), \sigma^2\mathbf{I}_n). \tag{22}$$

As the resulting posterior density is also Gaussian, the conditional mean estimate $\hat{\mathbf{w}}_{k|k}$ is the mode of (22) and determined as follows:

$$\begin{aligned}\hat{\mathbf{w}}_{k|k} &= \arg\min_{\mathbf{w}_k}\big(-\log p(\mathbf{w}_k|D_k)\big) \\ &= \arg\min_{\mathbf{w}_k}\big(\sigma^2[\mathbf{w}_k - \hat{\mathbf{w}}_{k|k-1}]^T P_{k|k-1}^{-1}[\mathbf{w}_k - \hat{\mathbf{w}}_{k|k-1}] \\ &\quad +[\mathbf{d}_k - \mathbf{h}(.,.)]^T[\mathbf{d}_k - \mathbf{h}(.,.)]\big),\end{aligned} \tag{23}$$

where

- the second term on the right hand side of (23) is the usual sum of squared error incurred in function approximation;
- the first term can be thought of as a regularizer for the approximation error criterion; specifically, the regularizer term can be though to measure the discrepancy between two neural networks of same size assessed by a weighted Euclidean distance between their corresponding weight vectors $\mathbf{w}_k$ and $\hat{\mathbf{w}}_{k|k-1}$.

[1] Regularization is a well-known method for treating mathematically ill-posed problems. It has been applied successfully to several machine learning problems to avoid over-fitting.

Setting the Jacobian of the regularized sum of squared error cost function (23) to zero yields the explicit solution to the weight estimate of the form (compare (16)):

$$\hat{\mathbf{w}}_{k|k} = \hat{\mathbf{w}}_{k|k-1} + G_k(\mathbf{d}_k - \hat{\mathbf{d}}_{k|k-1}).$$

where the predicted network output

$$\hat{\mathbf{d}}_{k|k-1} = \mathbb{E}\big[\mathbf{h}(\mathbf{w}_k, \mathbf{u}_k)|D_{k-1}\big], \tag{24}$$

and the Kalman gain G_k can be computed in a number of ways depending on which Bayesian filter is employed. From (20), we may equivalently write the weight estimate

$$\hat{\mathbf{w}}_{k|k} = \hat{\mathbf{w}}_{k-1|k-1} + G_k(\mathbf{d}_k - \hat{\mathbf{d}}_{k|k-1}). \tag{25}$$

Next, we illustrate how each Bayesian filter operates in the context of supervised training.

4.2 Extended Kalman Filters

Theory of the EKF results from linearizations of nonlinear functions and then applying the Kalman filter theory to nonlinear, Gaussian systems. The linearization relies on use of the first-order Taylor series expansion evaluated at the current estimate. Though the EKF is popular for its simplicity, it is plagued by the following limitations: (i) It quickly diverges in 'highly' nonlinear systems, owing to its linear approximation (ii) The EKF often yields an underestimated error covariance as it does not account for a prior covariance matrix in its analytical linearization approach (iii) Finally, its application is limited to differentiable functions only.

To elaborate the concept, we consider how the EKF computes its predicted network output in the supervised training mode of the neural network. Here, we assume that the prior density of $\mathbf{w}$, $p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, I_w)$. In the sequel, we suppress arguments $\mathbf{d}_k$ and $\mathbf{u}_k$ of $\mathbf{h}()$ for notational simplicity. We write the EKF-estimated predicted network output

$$\begin{aligned}\hat{\mathbf{d}}_{k|k-1} &= \mathbb{E}[\mathbf{h}(\mathbf{w}_k)|D_{k-1}] \\ &\approx \mathbb{E}[\mathbf{h}(\mathbf{0}) + J(\mathbf{0})\mathbf{w}_k] \\ &= \mathbf{h}(\mathbf{0}).\end{aligned}$$

Here $J \in \mathbb{R}^{n\times w}$ denotes the Jacobian matrix of $\mathbf{h}$. Here, the EKF estimate does not utilize the second-order statistics. Moreover, we may equivalently obtain the predicted network output of the EKF as

$$\hat{\mathbf{d}}_{k|k-1} = \mathbb{E}[\mathbf{h}(\mathbf{w}_k)|D_{k-1}] \approx \mathbf{h}(\mathbf{0}) = \mathbf{h}(\mathbb{E}(\mathbf{w}_k|D_{k-1})),$$

which implies that the EKF estimate is exact for linear functions, and may be highly biased otherwise. As a solution to mitigate bias errors of the EKF, the CDKF is discussed next.

4.3 Central Difference Kalman Filters

The CDKF is, in principle, comparable to the second-order EKF. That is, the nonlinear functions of the dynamic system are approximated by the second-order Taylor series expansion, in which the Jacobians and Hessians are replaced by central difference approximations. Use of the CDKF for training neural networks has had successes recently for the following two main reasons [19]: (i) It is derivative-free (ii) The square-root version of the original CDKF ensures a smooth continuous operation during training.[2]

To elaborate its operation, we write the steps involved in computing the predicted network output as follows:
(i) We write the second-order Taylor series expansion of $\mathbf{h}(\mathbf{w}_k)$ about $\mathbf{0}$:

$$\mathbf{h}(\mathbf{w}_k) \approx \mathbf{h}(\mathbf{0}) + J(\mathbf{0})\mathbf{w}_k + \frac{1}{2}\sum_{i=1}^{n} \mathbf{w}_k^T H_i(\mathbf{0})\mathbf{w}_k \mathbf{e}_i, \tag{26}$$

where $H_i \in \mathbb{R}^{w\times w}$ denotes the the square symmetric Hessian matrix whose elements are various second-order partial derivatives of $\mathbf{h}_i(.), i = 1, 2 \ldots n$ and $\mathbf{e}_i \in \mathbb{R}^n$ is the i-th elementary column vector.

(ii) Taking expectation of (26) with respect to $\mathbf{w}_k$ yields

$$\hat{\mathbf{d}}_{k|k-1} = \mathbb{E}[\mathbf{h}(\mathbf{w}_k)|D_{k-1}] \approx \mathbf{h}(\mathbf{0}) + \frac{1}{2}\sum_{j=1}^{w} \mathbf{Tr}[H_i(\mathbf{0})]\mathbf{e}_i, \tag{27}$$

where **Tr** denotes the trace operator.

(iii) Finally, we replace the Hessian in (27) by the central difference approximation. For example, we approximate the j-th diagonal element of the Hessian

$$[H_i(\mathbf{0})]_{j,j} = \frac{\partial^2 \mathbf{h}_i}{\partial w_j^2}(\mathbf{0}) \approx \frac{\mathbf{h}_i(\triangle \mathbf{e}_j) - 2\mathbf{h}_i(\mathbf{0}) + \mathbf{h}_i(-\triangle \mathbf{e}_j)}{2\triangle}$$

where the step-size $\triangle$ is chosen to be $\triangle = \sqrt{3}$ for a Gaussian prior.

The CDKF yields a more accurate estimate than that of the EKF. Secondly, it operates on function evaluations, thus is extendable to non-differential functions. However, several assumptions are made in order to arrive at the CDKF equations. For example, (i) when computing the first two-order moments, namely the mean and covariance, the CDKF essentially takes two approximations: (i.a) Quadratic interpolating polynomials for nonlinear functions and (i.b) numerically approximative Jacobians and Hessians (ii) the choice of the step size $\triangle$ to be $\triangle = \sqrt{3}$ is not always guaranteed to be optimal in all problem settings and (iii) finally, the interpolating point set using the fixed step-size of $\triangle = \sqrt{3}$ irrespective of the state dimension w may fail to capture global properties of a nonlinear function (e.g., peaks and troughs); the reason is that the quadratic function approximation is accurate only in the vicinity of the current estimate [2].

[2] The square-root CDKF is sometimes referred to as nprKF, where 'npr' stands for the initials of the three authors of [14].

4.4 Cubature Kalman Filters

As opposed to functional approximations to nonlinear functions, the CKF takes a more accurate and mathematically straightforward approach. As described earlier, the CKF computes the predicted network output as follows:

$$\begin{aligned}\hat{\mathbf{d}}_{k|k-1} = \mathbb{E}[\mathbf{h}(\mathbf{w}_k)|D_{k-1}] &= \int_{\mathbb{R}^w} \mathbf{h}(\mathbf{w})\mathcal{N}(\mathbf{w};\mathbf{0},I_w)d\mathbf{w} \\ &\approx \frac{1}{2w}\sum_{i=1}^{2w}\mathbf{h}(\xi_i),\end{aligned}$$

where cubature points ξ_i take a similar form as in Subsection 3.3

Remark. So far, we have discussed how each Bayesian filter computes the predicted network output. We may also extend the similar procedure to compute the covariance of the predicted error. The predicted error covariance, which is also called *innovation covariance*, essentially determines the accuracy of the Kalman gain. Because the CKF is proven to retain the information about the first two order moments of a nonlinearly transformed variable completely, we may expect the CKF to update the weights more accurately than other two Bayesian filters.

5 Practical Implementation

In this section, we describe a number of techniques of how to fine tune the CKF for better results.

5.1 Reducing Computational Complexity

As described earlier, the Bayesian filters implicitly minimizes the sum of squared error. To elucidate the relationship of the sum of squared error in (23) and the measurement function, we rewrite (2) as

$$\mathbf{0} = \big(\mathbf{d}_k - \mathbf{h}(\mathbf{w}_k, \mathbf{x}_k)\big) + \mathbf{r}_k, \tag{28}$$

where the output (measurement) is forced to take $\mathbf{0}$. Typically we encounter vector-valued measurements in n-class classification problems. In this case, we may replace the vector-valued measurement function by a scalar-valued measurement function:

$$0 = \tilde{h}(\mathbf{w}_k, \mathbf{x}_k, \mathbf{t}_k) + r_k, \tag{29}$$

where

$$\tilde{h}(.,.,.) = \sqrt{\sum_{i=1}^{n}\big(\mathbf{d}_k^{(i)} - \mathbf{h}^{(i)}(.,.,.)\big)^2}.$$

The above reformulation appears reasonable because the Bayesian filter minimizes the same sum of squared error. Also, the independency among output labels does not alter the *degree of nonlinearity* from the point of view of any single weight-variable. Finally, when any one of the outputs corresponding to the appropriate class is given, the other outputs carry little information. Based on these observations, we expect the Bayesian filter using (29) to yield a weight estimate similarly to that using (2). The similarity has also been verified in a number of simulations. The scalar reformulation offers the following benefits: (i) It reduces the computational complexity of the Bayesian filter based-training significantly (ii) It improves the numerically stability of the Bayesian filter. For example, when computing the Kalman gain, we replace the inversion of the innovation covariance matrix with a trivial scalar division (see Ch. 2, [9] also for a similar exposition in the EKF setup).

5.2 Fitting Various Cost Functions

In this subsection, we consider how to incorporate cost criteria other than the sum of squared error into the state-space model. To fit a given cost function, which may be non-differentiable, into the state-space model, we closely follow the idea of reformulating the measurement equation as explained through a couple of examples below.

Example 1: We consider the cross-entropic cost function that has been found to be appropriate for an n-class pattern-classification problem. When the classifier network uses the softmax output layer and the binary encoding for its outputs, we write the cross-entropic cost function

$$J_k = -\sum_{i=1}^{n} \mathbf{d}_k^{(i)} \log \mathbf{h}^{(i)}(\mathbf{w}_k, \mathbf{x}_k).$$

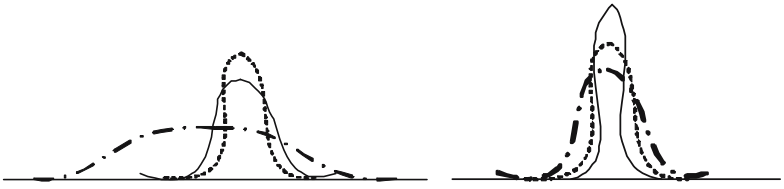
Correspondingly, we write the reformulated measurement function to be the square-root of J_k. That is, we write the measurement equation to take the form:

$$0 = \sqrt{-\sum_{i=1}^{n} \mathbf{d}_k^{(i)} \log \mathbf{h}^{(i)}(\mathbf{w}_k, \mathbf{x}_k)} + r_k. \tag{30}$$

Example 2: Here we consider the cost function to be the Minskowski's p-power metric. In this case, we write the reformulated measurement equation:

$$0 = \sqrt{\Big(\sum_{i=1}^{n} |\mathbf{d}_k^{(i)} - \mathbf{h}^{(i)}(\mathbf{w}_k, \mathbf{x}_k)|^p\Big)} + r_k. \tag{31}$$

Setting $p = 2$ in (31) yields the usual sum of squared error cost function. Moreover, the CKF and the CDKF can still be utilized when $p = 1$ (absolute error or $\mathcal{L}_1$-norm), whereas the EKF fails due to the unavailability of analytic gradient.



(a) At time $k = 1$ (b) At time $k = 1$

Fig. 3. Posterior (solid line) resulting from prior (dash-dot line) combined with likelihood function (dotted line) at two consecutive time steps

5.3 Choosing Hyperparameters

Both the accuracy of weight estimate and the rate of convergence of the CKF crucially hinge on the right choice of statistics of hyperparameters, namely, noise models and weight priors. Specifically, we assume that the given training data set is normalized and all the models are zero-mean Gaussian. In the Bayesian filtering framework, the assumption of Gaussianity allows to obtain a tractable solution. In this case, we are left to specify right covariances only. Next, we describe simple methods to compute them and explain why they work effectively.

Process noise covariance. Injecting process noise artificially helps escape poor local minima of the error surface. However, as the training progresses, the process noise covariance Q_k needs to be reduced systematically. We use a mathematical relationship of the form $Q_{k-1} = (\frac{1}{\lambda} - 1)P_{k-1|k-1}$, where $\lambda \in (0, 1]$ is called the the forgetting factor [28]. This allows to put exponentially decaying weight on past data. Consequently, for the linear process equation (20), we get the predicted error covariance $P_{k|k-1}$ to be $P_{k|k-1} = \frac{1}{\lambda}P_{k-1|k-1}$; this suggests that we get $P_{k|k-1} > P_{k-1|k-1}$ when predicting a new weight estimate before the receipt of a new measurement. Typically, the choice of λ being slightly less than unity works well for many problems.

Measurement noise covariance. Suppose we are given a set of clean training data set embedding the information about the unknown weight vector. Hence, it is reasonable to assume that the likelihood function is 'highly' peaked. We choose a fixed measurement noise covariance of the form $R_k = \sigma^2 I_n$. However, the choice may vary depending on the quality of the training data.

Initial error covariance. To choose this hyperparameter, we use the *principle of stable estimation*, which suggests to choose the prior density to be relatively more flat than the likelihood function in the case of uninformative prior [16]. The reason is that the shape of the posterior is forced to closely follow the shape of the likelihood rather than that of uninformative prior. We may choose the initial error covariance $P_{0|0}$ to be a diagonal matrix, with diagonal entries being one/two order higher than σ^2. The choice of initial covariance seems contrary to the Bayesian perspective, which is essentially built around the principle that *Priors rule; posteriors*

just follow around them. As training progresses however, the prior becomes more influential and decides the shape of the posterior (see Fig. 3).

5.4 Improving Stability: Square-Root Formulation

The two basic properties of an error covariance matrix are (i) symmetry and (ii) positive definiteness. In practice, due to finite word-length arithmetics, these two properties are often lost. Specifically, the positive definiteness of the posterior covariance, which is computed as the difference between two positive definite matrices is likely to be lost (see (17)). The conflict between theoretical properties and practical values may lead to an unstable behavior of a Bayesian filter. Importantly, the loss of the positive definiteness may be more hazardous as it stops the CKF to run continuously. The obvious question in this case is: how do we avoid this undesirable situation so as to assure a continuous stable operation of the CKF? The answer lies in a mathematically principled approach called square-root filtering.

The square-root version of the cubature Kalman filter hereafter called *square-root cubature Kalman filter* (SCKF), essentially propagates a square-root factor of the covariance matrix. To develop the SCKF for training neural networks, we use matrix factorizations. To elaborate further, we write the CKF-estimated covariance $P \in \mathbb{R}^{w \times w}$ of the form:

$$P = AA^T, \tag{32}$$

where $A \in \mathbb{R}^{w \times l}$ with $l > w$ is a fat matrix. In the square-root filtering, we are required to compute a square-root factor of P of dimension $(w \times w)$ without explicitly taking the square-root of P owing to numerical instabilities. Applying the QR decomposition on A^T, we get

$$P = AA^T = R^T Q^T Q R = R^T R = BB^T,$$

where the desired square-root factor is a lower triangular matrix $B = R^T$. Hereafter we call this procedure *triangularization*. Note that in computing the square-root covariance B, we discard the orthogonal matrix Q and exploit the upper triangular matrix R only. Since B is triangular, its sparseness provides efficient computation and reduced storage space. In terms of computational complexity, the SCKF requires $O(w^3)$ *flops*, which comes from the use of a matrix triangularization.

6 Experimental Results

In this section, we report our findings of the experiment dealing with Bayesian filter-trained RNNs. The trained RNNs are employed to predict the chaotic Mackey-Glass time-series. We use the CKF as described in Appendix A, the CDKF and the EKF as training algorithms.

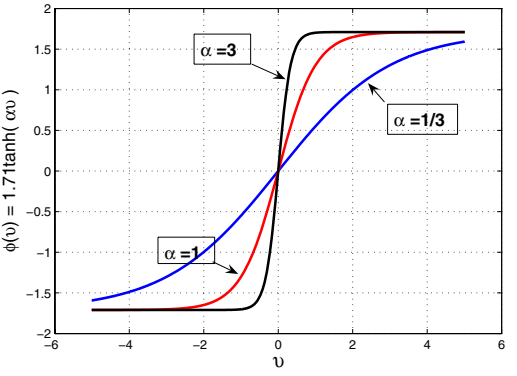


Fig. 4. Effect of α on the shape of the activation function $\varphi(\upsilon) = 1.71\tanh(\alpha\upsilon)$

Chaotic Mackey-Glass System. The Mackey-Glass equation is often used to model the production of white-blood cells in Leukemia patients and given by the delay differential equation:

$$\frac{dx_t}{dt} = 0.1x_t + \frac{0.2x_{t-\Delta}}{1 + x_{t-\Delta}^{10}}, \tag{33}$$

where the delay $\Delta = 30$. To sample the time-series at discrete time steps, we numerically integrated (33) using the forth-order Runge-Kutta method with a sampling period of $T = 6$ s, and the initial condition $x_t = 0.9$, for $0 \leq t \leq \Delta$. Given a chaotic system, it is known that the next data sample $x_{k+\tau}$ can be predicted from a properly chosen time-series $\{x_k, x_{k-\tau} \ldots x_{k-[d_E-2]\tau}, x_{k-[d_E-1]\tau}\}$, where d_E and τ are called the embedding dimension and the embedding delay, respectively. For the chaotic Mackey-Glass system, d_E and τ were chosen to be seven and one, respectively.

RNN Architecture. We used the Bayesian filter-trained RNNs to predict the chaotic Mackey-Glass time-series data. We chose the RNN to have seven inputs representing an embedding of the observed time-series, one output, and one self-recurrent hidden layer with five neurons. Hence, the RNN has a total of 71 connecting weights (bias included). The output neuron uses a linear activation function, whereas all the hidden neurons use the hyperbolic tangent function of the form

$$\varphi(\upsilon) = 1.71 \tanh(\alpha\upsilon),$$

where α was assumed to take values ranging from 1/3 - 3. As shown in Fig. 4, the hyperbolic tangent function is 'mildly' nonlinear (that is, close to a linear function) around its origin when $\alpha = 1/3$. Its nonlinearity increases with α, and behaves closely similar to a switch when $\alpha = 3$.

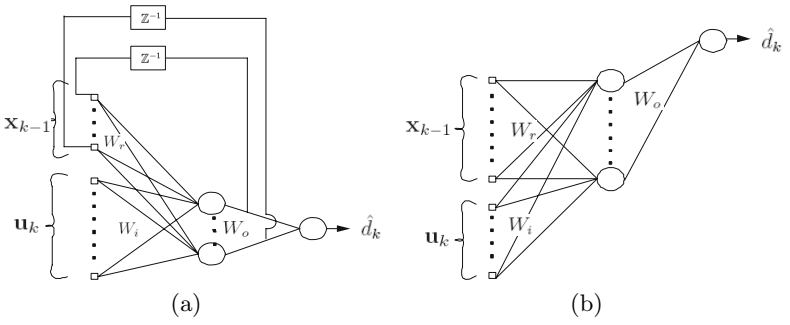


Fig. 5. Schematic diagrams (a). Original RNN (b). Unfolded RNN of unity truncation depth.

State-Space Model. We represent the RNN architecture following the state-space model:

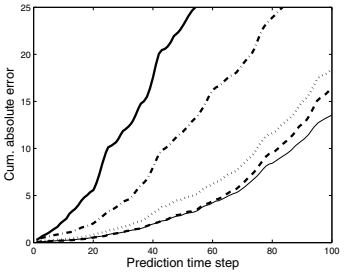
$$\begin{aligned}\mathbf{w}_k &= \mathbf{w}_{k-1} + \mathbf{q}_{k-1}\\ 0 &= \Big(d_k - W_o\varphi\big(W_r\mathbf{x}_{k-1} + W_i\mathbf{u}_k\big)\Big) + r_k.\end{aligned}$$

Here φ_o denotes the softmax function of the output layer, whereas φ_i and φ_r denote the hyperbolic tangent functions of the input layer and recurrent layer, respectively; W_i, W_r and W_o are input, recurrent and output weight matrices of appropriate dimensions; the weight vector $\mathbf{w}_k$ is obtained by grouping elements from W_i, W_r and W_o in 'some' orderly fashion.

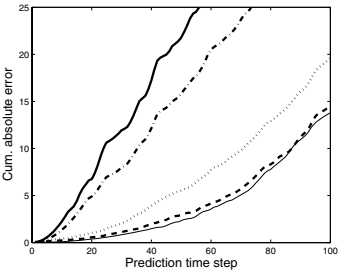
Data. We obtained the chaotic time series of length 1000, of which the first half was used for training and the rest for testing. To train the RNN using the CKF, we used 10 epochs per run. Each epoch was obtained from a 107 time-step long subsequence, starting from a randomly selected point. That is, each epoch consists of 100 examples gleaned by sliding a window of length eight over the subsequence. The weights were initialized to zero-mean Gaussian with a diagonal covariance of $0.5I_w$; We made Q_{k-1} to decay such that $Q_{k-1} = (\frac{1}{\lambda} - 1)P_{k-1|k-1}$ with $\lambda = 0.9995$. We fixed R_k to be $R_k = 5 \times 10^{-3}$ across all entire epochs; the output of the hidden layer (state) of the RNN at $t = 0$, $\mathbf{x}_0$ was assumed to be zero.

As opposed to the CKF relying on integration, the EKF and the CDKF use gradient information, which necessitate the use of the truncated backpropagation through time method. The truncation depths of longer than unity were tried; surprisingly, a length of unity was found to be sufficient in this experimental setup (see Fig. 5).

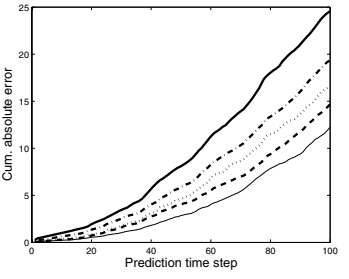
Performance Metric. During the testing phase, we initialized RNNs with 20 time-step long test sequence and allowed them to run autonomously using their own output for the next 100 steps. To compare the performance of CKF-trained RNNs against the CDKF, and EKF-trained RNNs fairly, we made 50 independent training runs for each value of α. As a performance metric, we used the ensemble-averaged cumulative absolute error



(a) EKF-trained RNN.



(b) CDKF-trained RNN.



(c) CKF-trained RNN.

Fig. 6. Ensemble-averaged cumulative absolute error curves during the autonomous prediction when $\alpha = 1/3$ (solid-thin), 2/3 (dashed), 1 (dotted), 2 (dash-dot), and 3 (solid-thick)

$$e_k = \frac{1}{50} \sum_{r=1}^{50} \sum_{i=1}^{k} |d_i^{(r)} - \hat{d}_i^{(r)}|; \quad k = 1, 2, \ldots 100.$$

The long-term accumulative prediction error is increasing exponentially in time k for the following reasons: (i) the chaotic systems are known to be sensitive even to a slight perturbation in their present state [15] (ii) the prediction error is amplified at each time step due to the closed loop structure. From Figs. 6(a) and 6(b), we see that the EKF and CDKF-trained RNNs break down at $\alpha = 2$ and beyond. The CKF-trained RNN performs reasonably well even when $\alpha = 3$ at which the hyperbolic tangent function is 'severely' nonlinear (Fig. 6(c)). The reason is that the CKF tends to locate a better local minimum of the cost function in the weight space than the EKF or the CDKF. The successful solution to estimating more accurate weights in a highly nonlinear setup studied herein is a convincing testimony to the superiority of the CKF as a training algorithm.

7 Concluding Remarks

In this paper, we introduced the cubature Kalman filter (CKF) for training RNNs for the first time in the machine learning literature. With the cubature rule at our disposal, the principle behind the CKF can be confined to linear filtering

theory. Despite algorithmic differences, the approximate Bayesian filters exemplified by the EKF, the CDKF and the CKF, share a common feature: *they all are second-order optimizers minimizing the regularized sum of square error cost function*. Specifically, it has already been proved that the EKF is equivalent to the Newton's method whereas the backpropagation is equivalent to a degenerate form of the EKF in [21], [23]. Hence, the Bayesian filters provide more accurate solution and rapid convergence than the gradient method. A significant performance boost of Bayesian filters over the gradient descent method results from computing (i) the predicted network output ([24]) and (ii) the Kalman gain G_k joining the prediction error to the latest weight estimate. Of course, a comparison of Newton's method and the weight update of a Bayesian filter suggests that the Kalman gain encompasses a stochastic counterpart of a scaled inverse Hessian of the cost function. The Kalman gain can also be thought of as a collection of learning rates chosen optimally for each neuron of the network.

Specifically, CKF-trained RNNs seem to outperform other Bayesian filter-trained RNNs. Apart from its more accurate solution to the weight vector, the CKF may be preferred for the following reasons:

1. The CKF does not require analytical derivatives.
2. The CKF includes only the forward pass as opposed to forward and backward passes as in the gradient-based training algorithms. Hence, we do not keep track of information local to each unit.
3. In the CKF-based training of a RNN, we only consider the unfolded architecture of unity truncation depth. Hence, it is extremely easy to code and implement the training algorithm for a RNN.
4. Last but by no means the least, we mention the benefit of the fast square-root algorithm tailored specifically for training neural networks. It ensures a continues smooth training operation.

Acknowledgement

The authors would like to thank the natural sciences and engineering research council (NSERC) of Canada for financially supporting this project.

References

1. Anderson, B.D.O., Moore, J.B.: Optimal Filtering. Prentice Hall, New Jersey (1979)
2. Arasaratnam, I., Haykin, S.: Cubature Kalman Filters. IEEE Trans. Automatic Cont. (2008) (under 3rd review)
3. Bengio, Y., Simard, P., Frasconi, P.: Learning Long-term Dependencies with Gradient Descent is Difficult. IEEE Trans. Neural Netw. 5(2), 157–166 (1994)
4. Cappe, O., Godsil, S.J., Moulines, E.: An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. Proc. IEEE 95(5) (2007)
5. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. IEE Proc. -F. 140, 107–113 (1993)

6. Hanson, S.J., Burr, D.J.: Minkowski-r Back-Propagation: Learning in Connectionist Models with Non-Euclidian Error Signals. In: Anderson, D. (ed.) Neural inform. proces. sys., pp. 348–357. American Inst. of Physics, NY (1988)
7. Haykin, S., Li, L.: Nonlinear Adaptive Prediction of Nonstationary Signals. IEEE Trans. Signal Process 43(2), 526–535 (1995)
8. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, New Jersey (1999)
9. Haykin, S. (ed.): Kalman Filtering and Neural Networks. Wiley, New York (2001)
10. Julier, S.J., Ulhmann, J.K., Durrant-Whyte, H.F.: A New Method for Nonlinear Transformation of Means and Covariances in Filters and Estimators. IEEE Trans. Automatic Cont. 45, 472–482 (2000)
11. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proc. IEEE 86(11), 2278–2324 (1998)
12. Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer, Heidelberg (2001)
13. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Trans. Neural Netw. 1(1), 4–27 (1990)
14. Nørgaard, M.N., Poulsen, N.K., Ravn, O.: New Developments in State Estimation of Nonlinear Systems. Automatica 36, 1627–1638 (2000)
15. Ott, E.: Chaos in Dynamical Systems, 2nd edn. Cambridge Univ. Press, Cambridge (2002)
16. Peterka, V.: Bayesian Approach to System Identification. In: Eykhoff, P. (ed.) Trends and progress in system identification, pp. 239–304. Pergamon Press, Oxford (1981)
17. Pineda, F.: Generalization of Backpropagation to Recurrent Neural Networks. Physical Review Letters 59(19) (1987)
18. Plumer, E.S.: Training Neural Networks Using Sequential Extened Kalman Filters. In: Proc. World Congress on Neural Netw., Washigton, DC, pp. I-764–I-769 (1995)
19. Prokhorov, D.: Training Recurrent Neurocontrollers for Robustness with Derivative-Free Kalman Filter. IEEE Trans. Neural Netw. 17(6), 1606–1616 (2006)
20. Puskorious, G., Feldkamp, L.: Neurocontrol of Nonlinear Dynamical Systems with Kalman Filter Trained Neural Networks. IEEE Trans. Neural Netw. 5(2) (1994)
21. Ruck, D.W., Rogers, S.K., Kabrisky, M., Maybeck, P., Oxle, M.E.: Comparative Analysis of Backpropgation and the Extended Kalman Filter for Training Multilayer Perceptrons. IEEE Trans. Patt. Anal. & Mach. Intell. 14(6), 686–691 (1992)
22. Sarangapani, J.: Neural Network Control of Nonlinear Discrete-Time Systems. CRC Press, Florida (2006)
23. Schottky, B., Saad, D.: Statistical Mechanics of EKF Learning in Neural Networks. J. Physics A 32(9), 1605–1621 (1999)
24. Shah, S., Palmieri, F., Datum, M.: Optimal Filtering Algorithms for Fast Learning in Feedforward Neural Networks. Neural Net. 5(5), 779–787 (1992)
25. Sum, J., Leung, C., Young, G.H., Kan, W.: On the Kalman Filtering Method in Neural Network Training and Pruning. IEEE Trans. Neural Netw. 10(1) (1999)
26. Yamada, T., Yabuta, T.: Dynamic System Identification Using Neural Networks. IEEE Trans. Systems, Man and Cyber. 23(1), 204–211 (1993)
27. Werbos, P.: Backpropagation Through Time: What It Does and How to Do It. Proc. IEEE 78(10), 1550–1560 (1990)
28. West, M., Harrison, J.: Bayesian Forecasting and Dynamic Linear Models. Springer, Heidelberg (1996)

Appendix A: Square-Root Cubature Kalman Filter

State-space model: In the SCKF algorithm outlined below, we assume the following reformulated state-space model:

$$\begin{aligned}\mathbf{w}_k &= \mathbf{w}_{k-1} + \mathbf{q}_{k-1} \\ 0 &= \tilde{h}(\mathbf{w}_k, \mathbf{u}_k, \mathbf{d}_k) + r_k,\end{aligned}$$

where $\tilde{h}(\mathbf{w}_k, \mathbf{u}_k, \mathbf{d}_k) = \sqrt{\sum_{i=1}^{n} \left(\mathbf{d}_k^{(i)} - \mathbf{h}^{(i)}(\mathbf{w}_k, \mathbf{u}_k)\right)^2}$ for the cost function being the sum of squared error; $\mathbf{q}_{k-1}$ and r_k are independent, zero-mean Gaussian noise sequences with covariances Q_{k-1} and σ^2; Q_{k-1} is annealed using the adaptive strategy with a forgetting factor λ.

Initial Assumption: At time $(k-1)$, we are given the posterior density

$$p(\mathbf{w}_{k-1}|D_{k-1}) = \mathcal{N}(\hat{\mathbf{w}}_{k-1|k-1}, S_{k-1|k-1}S_{k-1|k-1}^T).$$

SCKF Algorithm from time $(k-1)$ to k: Compute

1. Cubature points

$$\mathscr{W}_i = \hat{\mathbf{w}}_{k-1|k-1} + \frac{1}{\sqrt{\lambda}} S_{k-1|k-1}\xi_i \quad i = 1, 2, \ldots, 2w. \tag{34}$$

   where the cubature points $\{\xi_i\}$ are defined in Subsection 3.3.
2. Propagated cubature points

$$\mathscr{D}_i = \tilde{h}(\mathscr{W}_i, \mathbf{u}_k, \mathbf{d}_k) \quad i = 1, 2, \ldots, 2w. \tag{35}$$

3. Predicted network output

$$\hat{d}_{k|k-1} = \frac{1}{2w} \sum_{i=1}^{2w} \mathscr{D}_i. \tag{36}$$

   Hence, the innovation is $(-\hat{d}_{k|k-1})$.
4. Innovation variance

$$\sigma_{d,k|k-1}^2 = \mathscr{D}\mathscr{D}^T + \sigma^2, \tag{37}$$

   where the weighted, centered row vector

$$\mathscr{D} = \frac{1}{\sqrt{2w}}[\mathscr{D}_1 - \hat{d}_{k|k-1} \quad \mathscr{D}_2 - \hat{d}_{k|k-1} \ldots \mathscr{D}_{2w} - \hat{d}_{k|k-1}]. \tag{38}$$

5. Cross-covariance

$$P_{wd,k|k-1} = \mathscr{W}\mathscr{D}^T, \tag{39}$$

   where the weighted, centered matrix

$$\mathscr{W} = \frac{1}{\sqrt{2w}}[\mathscr{W}_1 - \hat{\mathbf{w}}_{k-1|k-1} \quad \mathscr{W}_2 - \hat{\mathbf{w}}_{k-1|k-1} \ldots \mathscr{W}_{2w} - \hat{\mathbf{w}}_{k-1|k-1}]. \tag{40}$$

6. Kalman gain

$$G_k = P_{wd,k|k-1}/\sigma^2_{d,k|k-1}. \tag{41}$$

7. Updated weight

$$\hat{\mathbf{w}}_{k|k} = \hat{\mathbf{w}}_{k-1|k-1} - G_k \hat{d}_{k|k-1}. \tag{42}$$

8. Square-root factor of the posterior error covariance

$$S_{k|k} = \mathbf{Tria}([\mathscr{W} - G_k \mathscr{D} \quad \sigma G_k]), \tag{43}$$

where we denote the triangularization algorithm as $B = \mathbf{Tria}(A)$ with the output B being a lower triangular matrix. The matrices A and B are related as follows: Let the upper triangular matrix R be obtained from the QR decomposition on A^T; then $B = R^T$.

QuBIS: An (In)complete Solver for Quantified Boolean Formulas

Luca Pulina and Armando Tacchella*

DIST, Università di Genova, Viale Causa, 13 – 16145 Genova, Italy
Luca.Pulina@unige.it - Armando.Tacchella@unige.it

Abstract. In this paper we introduce QuBIS an (in)complete solver for quantified Boolean formulas (QBFs). The particularity of QuBIS is that it is not inherently incomplete, but it has the ability to surrender upon realizing that its deduction mechanism is becoming ineffective. Whenever this happens, QuBIS outputs a partial result which can be fed to a complete QBF solver for further processing. As our experiments show, not only QuBIS is competitive as an incomplete solver, but providing the output of QuBIS as an input to complete solvers can boost their performances on several instances.

1 Introduction

The problem of evaluating quantified Boolean formulas (QBFs) is one of the cornerstones of Complexity Theory. In its most general form, it is the prototypical PSPACE-complete problem, also known as QSAT [19]. Introducing limitations on the number and the placement of alternating quantifiers, QSAT is complete for each class in the polynomial hierarchy (see, e.g., [16]). Therefore, QBFs can be seen as a low-level language in which high-level descriptions of several hard combinatorial problems can be encoded to find a solution by means of a QBF solver, and it has been shown that QBFs can provide compact propositional encodings in many automated reasoning tasks (see, e.g., [11,1,6]). The interest in QSAT is also witnessed by a number of QBF encodings and solvers which are made publicly available (see [8]), and by the presence of an annual competition of QBF solvers (QBFEVAL) [13].

In this paper we introduce QuBIS, an (in)complete solver for QBFs. The particularity of QuBIS is that it is not inherently incomplete, but it has the ability to surrender upon realizing that its deduction mechanism is becoming ineffective. Whenever this happens, QuBIS outputs a partial result which can be fed to a complete QBF solver for further processing. QuBIS is based on *Q-resolution* defined in [12] as an operation extending standard propositional resolution to QBFs represented in prenex clausal form. In particular, QuBIS uses Q-resolution to eliminate existentially quantified variables, yielding an extension of the well-known Davis-Putnam [5] procedure for propositional satisfiability (SAT). In this aspect, QuBIS is similar to the state-of-the-art solvers

* The authors wish to thank MIUR for its financial support, and the reviewers who helped to improve the original manuscript.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 34–43, 2008.

QMRes [15] and quantor [3]. However, our approach differs both from QMRes – we do not use symbolic data structures – and from quantor – we never expand universal variables. Furthermore, the policy for selecting variables to eliminate is very different from the ones used in quantor and QMRes. In this regard, QuBIS is more related to the Bounded Directional Resolution (BDR) algorithm presented in [17]. The main difference between BDR and QuBIS is that our solver uses dynamic, rather than static, reordering of variables. However, both QuBIS and BDR leverage information on the structure of the formula to solve (or preprocess) their input.

Our experiments show that QuBIS – used as an incomplete solver – is able to solve non-trivial QBFs, and that it is also competitive with WalkQSAT [10], the only (inherently) incomplete QBF solver that was proposed so far. Furthermore QuBIS – used as a preprocessor – will boost the performances of state-of-the-art complete QBF solvers on several QBF encodings. Noticeably, the improvement includes instances that are usually difficult for most solvers, or where solvers exhibit very different behaviors depending on their internal algorithms. In this sense, we show that QuBIS somehow equalizes the differences between solver algorithms.

The paper is structured as follows. In Section 2 we introduce the notation and the basic definitions used in the paper. In Section 3 we describe QuBIS in some detail, while in Section 4 we show that it can effectively compete with WalkQSAT and also improve the performances of state-of-the-art QBF solvers. We conclude in Section 5 with some final remarks and the related work.

2 Preliminaries

In this section we consider the definition of QBFs and their satisfiability as given in the literature of QBF decision procedures (see, e.g., [9,2,3]).

A *variable* is an element of a set P of propositional letters and a *literal* is a variable or the negation thereof. We denote with $|l|$ the variable occurring in the literal l, and with $\overline{l}$ the *complement* of l, i.e., $\neg l$ if l is a variable and $|l|$ otherwise. A literal is *positive* if $|l| = l$ and *negative* otherwise. A *clause* C is an n-ary ($n \geq 0$) disjunction of literals such that, for any two distinct disjuncts l, l' in C, it is not the case that $|l| = |l'|$. A *propositional formula* is a k-ary ($k \geq 0$) conjunction of clauses. A *quantified Boolean formula* is an expression of the form

$$Q_1 z_1 \dots Q_n z_n \Phi \tag{1}$$

where, for each $1 \leq i \leq n$, z_i is a variable, Q_i is either an existential quantifier $Q_i = \exists$ or a universal one $Q_i = \forall$, and Φ is a propositional formula in the variables $\{z_1, \dots, z_n\}$. The expression $Q_1 z_1 \dots Q_n z_n$ is the *prefix* and Φ is the *matrix* of (1). A literal l is *existential* if $|l| = z_i$ for some $1 \leq i \leq n$ and $\exists z_i$ belongs to the prefix of (1), and it is *universal* otherwise. For example, the following expression is a QBF:

$$\begin{aligned}\forall y_1 \exists x_1 \forall y_2 \exists x_2 \exists x_3 ((y_1 \vee y_2 \vee x_2) \wedge (y_1 \vee \neg y_2 \vee \neg x_2 \vee \neg x_3) \wedge \\ (y_1 \vee \neg x_2 \vee x_3) \wedge (\neg y_1 \vee x_1 \vee x_3) \wedge \\ (\neg y_1 \vee y_2 \vee x_2) \wedge (\neg y_1 \vee y_2 \vee \neg x_2) \wedge \\ (\neg y_1 \vee \neg x_1 \vee \neg y_2 \vee \neg x_3) \wedge \\ (\neg x_2 \vee \neg x_3))\end{aligned} \tag{2}$$

The semantics of a QBF φ can be defined recursively as follows. A QBF clause is *contradictory* exactly when it does not contain existential literals. If the matrix of φ contains a contradictory clause then φ is false. If the matrix of φ has no conjuncts then φ is true. If $\varphi = Qz\psi$ is a QBF and l is a literal, we define φ_l as the QBF obtained from ψ by removing all the conjuncts in which l occurs and removing $\overline{l}$ from the others. Then we have two cases. If φ is $\exists z\psi$, then φ is true if and only if φ_z or $\varphi_{\neg z}$ are true. If φ is $\forall z\psi$, then φ is true if and only if φ_z and $\varphi_{\neg z}$ are true. The QBF satisfiability problem (QSAT) is deciding whether a given formula is true or false. It is easy to see that if φ is a QBF without universal quantifiers, solving QSAT is the same as solving propositional satisfiability (SAT).

3 The Design of QuBIS

In this section we introduce QuBIS, an (in)complete solver for quantified Boolean formulas. QuBIS is incomplete in that, given an input QBF φ, it may either halt producing a satisfiability result, or halt producing another QBF φ' having, in most cases, a smaller number of variables and longer clauses in the matrix . QuBIS is based on *Q-resolution* defined in [12] as an operation among clauses of a QBF. In particular, given two (non-tautological) clauses $P \vee x$ and $R \vee \neg x$, where P and R are disjunctions of literals, the clause $P \vee R$ can be derived by Q-resolution subject to the constraints that (i) x is an existential variable, and (ii) P and R do not share any variable z such that $\neg z$ (resp. z) occurs in P and z (resp. $\neg z$) occurs in R. QuBIS uses Q-resolution to perform *variable elimination* on existential variables defined, e.g., in [3], as the operation whereby, given a QBF $Q_1 z_1 Q_2 z_2 \ldots \exists x \Phi$, the variable x can be resolved away by performing all resolutions on x, adding the resolvents to the matrix Φ and removing from Φ all the clauses containing x. For example, given the QBF (2):

$$\begin{aligned}\forall y_1 \exists x_1 \forall y_2 \exists x_2 \exists x_3 ((y_1 \vee y_2 \vee x_2)_1 \wedge (y_1 \vee \neg y_2 \vee \neg x_2 \vee \neg x_3)_2 \wedge \\ (y_1 \vee \neg x_2 \vee x_3)_3 \wedge (\neg y_1 \vee x_1 \vee x_3)_4 \wedge \\ (\neg y_1 \vee y_2 \vee x_2)_5 \wedge (\neg y_1 \vee y_2 \vee \neg x_2)_6 \wedge \\ (\neg y_1 \vee \neg x_1 \vee \neg y_2 \vee \neg x_3)_7 \wedge \\ (\neg x_2 \vee \neg x_3)_8)\end{aligned}$$

resolving away the variable x_3 yields the QBF:

$$\begin{aligned}\forall y_1 \exists x_1 \forall y_2 \exists x_2 ((y_1 \vee y_2 \vee x_2)_1 \wedge (y_1 \vee \neg y_2 \vee \neg x_2)_{2,3} \wedge \\ (y_1 \vee x_1 \vee \neg y_2 \vee \neg x_2)_{2,4} \wedge (\neg y_1 \vee y_2 \vee x_2)_5 \wedge \\ (\neg y_1 \vee y_2 \vee \neg x_2)_6 \wedge (y_1 \vee \neg x_2)_{8,3} \wedge \\ (\neg y_1 \vee x_1 \vee \neg x_2)_{8,4})\end{aligned} \tag{3}$$

wherein the resolvents $7, 3$ and $7, 4$ have not been considered since they do not fulfill condition (ii) above. Universal variables can be eliminated simply by deleting them once they have the highest prefix level. More precisely, given a matrix Φ, let $\Phi_{/z}$ be the same matrix whereby all the occurrences of z have been deleted. The QBF $Q_1 z_1 Q_2 z_2 \ldots \forall y \Phi$ is true exactly when the QBF $Q_1 z_1 Q_2 z_2 \ldots \Phi_{/y}$ is true, so y can be eliminated safely. From the above, it immediately follows that variable elimination yields a (sound and complete) decision procedure for QSAT which is the extension of the Davis-Putnam [5] decision procedure for SAT (see, e.g., [15,3]).

QuBIS takes as input a QBF φ and two parameters:

- an integer *deg*, the maximum length that clauses can reach, and
- an integer *div*, the maximum value of *diversity*, a parameter defined in [17] as the product of the number of positive and negative occurrences of a variable.

The role of *deg* is thus to forbid clauses of excessive length to be generated, while the role of *div* is to bound the (worst case) number of resolvents generated when eliminating an existential variable. Intuitively, QuBIS eliminates variables until the input QBF can be declared true, false or when eliminating variables is bound to increase the size of the resulting QBF beyond some critical threshold.

More precisely, in QuBIS, a variable qualifies for elimination only if it has the highest level in the prefix of φ, and

- it is a universal variable, or
- it is an existential variable, it has degree no larger than *deg* and diversity no larger than *div*.

Universal variables are eliminated simply by deleting all their occurrences from the matrix of φ, while existential variables are resolved away. In both cases, the resulting QBF is given as argument to a recursive call of QuBIS. QuBIS terminates when one of the following conditions is satisfied:

1. the matrix of φ is empty – in which case the input QBF is true;
2. the matrix of φ contains a contradictory clause – in which case the input QBF is false;
3. there are no variables that qualify for elimination in φ – in which case φ is returned as output.

Therefore, QuBIS is a sound and complete decision procedure for the subclass of QBFs in which variables always qualify for elimination, while for all the other formulas QuBIS behaves like a preprocessor.

The implementation follows the algorithm outlined above with the following additional provisions:

- For efficiency reasons, data structures are updated destructively and recursion is replaced by iteration.
- When eliminating existential variables, priority is given to those occurring in *unit clauses*, i.e., clauses comprised of just one literal.

Table 1. Families of QBFs used to experiment with QuBIS

Family	QBFs	Description
`Adder`	16	Equivalence checking of partial implementations of circuits
`Chain`	4	Equivalence checking of partial implementations of circuits
`Connect4`	13	Encodings of the "Connect4" game into QBF
`Counter`	5	Model checking of counter circuits
`k`	82	Modal K formulas
`MutexP`	4	Equivalence checking of partial implementations of circuits
`Qshifter`	4	Barrel shifter problems
`s`	11	symbolic diameter evaluation of ISCAS89 circuits
`SzymanskiP`	9	Equivalence checking of partial implementations of circuits
`tipdiam`	30	Symbolic diameter evaluation of circuits

– Once all such variables are eliminated, the next existential variable is chosen among the remaining ones that qualify for elimination by ordering them for increasing degree, and breaking ties using minimum diversity.
– Forward subsumption checking is computed after elimination: new resolvents are not added if they are subsumed by some clause in the matrix.

4 Empirical Analysis

The purpose of this section is twofold. On one hand, we want to compare the performances of QuBIS with WalkQSAT which is, to the best of our knowledge, the only existing incomplete solver for QBFs. On the other hand, we want to show that most QBF solvers benefit from getting the output of an incomplete run of QuBIS rather than being fed the original QBF as input. All the experiments detailed in this section ran on a farm of 8 identical PCs, equipped with Intel Core 2 Duo 2.13 GHz with 4GB of RAM and running Debian/GNU Linux. In our experiments, all solvers were limited to 600 seconds of CPU time and 3GB of memory. The QBFs used throughout the experiments are from the last QBFEVAL wherein WalkQSAT participated (QBFEVAL'06 [14]). The selection was made considering QBF encodings only, i.e., excluding randomly generated formulas. Table 1 summarizes the families of QBFs used for our experiments. Further details can be found in [8].

In the first experiment we compare the performances of QuBIS with WalkQSAT. As described in Section 3, different settings of parameters *deg* and *div* can generate different behaviours of QuBIS. We run QuBIS with several settings and here we present two of them: the first one ($deg = 20$ and $div = 2000$) is preprocessing-oriented functions, while the second one ($deg = 40$ and $div = 4000$) is more solver-oriented. In the following, we will refer to these two settings as QuBIS$_p$ and QuBIS$_s$, respectively.

Table 2 shows the results of the above experiment. The first column contains the family name of QBFs, the second column ("N") denotes the number of formulas for each family, and they are followed by three groups of columns.

Table 2. Performances of $\text{QuBIS}_{p,s}$ and WalkQSAT on a selection of QBFs

Family	N	QuBIS_p		QuBIS_s		WalkQSAT	
		#	Time	#	Time	#	Time
`Adder`	16	–	–	–	–	–	–
`Chain`	4	4	0.14	4	0.14	1	516.60
`Connect4`	13	10	4.72	10	4.74	–	–
`Counter`	5	–	–	–	–	–	–
`k`	82	14	5.02	45	40.53	9	1341.35
`MutexP`	4	4	27.32	4	24.07	–	–
`Qshifter`	4	4	12.07	4	12.03	–	–
`s`	11	2	32.77	2	34.37	1	32.65
`SzymanskiP`	9	–	–	–	–	5	9.87
`tipdiam`	30	10	0.14	12	0.77	30	68.63
Total	178	48	82.18	81	116.65	46	1961.10

The first two groups are related to the settings of QuBIS, while the last one is related to WalkQSAT. For each group, the columns "#" and "Time" contain, respectively, the number of formulas solved and the cumulative CPU seconds. A dash on both columns means that the solver did not solve any formula. The last row ("Total") summarizes "#" and "Time" for all families.

Looking at Table 2, we can see that the performances of QuBIS and WalkQSAT can differ substantially across different families. In the case of `Connect4`, `MutexP`, and `Qshifter`, QuBIS is able to solve a relatively large number of formulas, while WalkQSAT solves none. Considering the `SzymanskiP` family instead, we see that WalkQSAT solves 5 formulas out of 9, while both settings of QuBIS solve none. The main motivation of this phenomenon is that the formulas solved by QuBIS tend to be simpler for solvers that use variable elimination, as QBFEVAL results of, e.g., quantor and QMRes clearly show. On the other hand, the family `SzymanskiP` contains instances that are easy for search-based solvers, which is exactly the main algorithm of WalkQSAT. Looking at families `k` and `tipdiam`, we can see a different behaviour between the different settings of QuBIS: QuBIS_s can solve 81 formulas out of 178, while QuBIS_p 48. In this case, QuBIS_s "aggressive" setting enables it to surrender later than QuBIS_p in the deduction process. While QuBIS_s setting usually pays off in terms of the number of problems solved, it is possible that the formula that it outputs when giving up will not be any easier to solve by a complete solver. From the experiment shown in Table 2 we can conclude that QuBIS_s is competitive with respect to WalkQSAT and that the characteristics of the two solvers complement each other.

The aim of the next experiment is to show how the performances of solvers using diverse reasoning algorithms can be improved using QuBIS. In particular, we consider QuBIS_p setting and the following QBF solvers:

QMRes, a symbolic implementation of variable elimination featuring multi-resolution, unit propagation, and heuristics to choose variables [15].

Table 3. Performances of a selection of QBF solver on a selection of QBFs with and without preprocessing by QuBIS

Family	N	QMRes				quantor				QuBE3.0			
		#	Time	#′	Time′	#	Time	#′	Time′	#	Time	#′	Time′
Adder	16	16	1061.41	16	713.41	4	20.78	4	34.08	1	1.80	2	16.56
Chain	4	4	6.37	4	0.14	4	0.14	4	0.14	–	–	4	0.14
Connect4	13	–	–	10	4.72	10	5.28	10	4.72	12	58.03	13	159.26
Counter	5	1	17.47	1	21.26	5	15.98	5	15.39	2	89.43	2	90.06
k	82	60	573.43	68	222.94	57	294.17	67	323.82	13	1113.72	32	906.39
MutexP	4	2	7.53	4	27.32	–	–	4	27.32	–	–	4	27.32
Qshifter	4	4	19.11	4	12.07	4	16.10	4	12.07	–	–	4	12.07
s	11	5	35.48	9	1428.78	11	468.18	11	2491.34	–	–	3	32.98
SzymanskiP	9	–	–	1	83.47	–	–	1	166.59	9	239.27	1	83.24
tipdiam	30	22	190.77	25	880.84	29	54.29	29	109.98	28	213.74	28	742.60
Total	178	114	1911.57	142	3394.95	124	874.92	139	3185.45	65	1715.99	93	2070.62

Family	N	QuBE6.1				sKizzo				yQuaffle			
		#	Time	#′	Time′	#	Time	#′	Time′	#	Time	#′	Time′
Adder	16	1	1.16	3	15.62	9	700.83	9	273.59	–	–	2	35.57
Chain	4	4	0.36	4	0.14	4	0.30	4	0.14	–	–	4	0.14
Connect4	13	13	206.98	13	70.76	11	114.66	12	631.29	12	232.82	12	134.87
Counter	5	2	115.86	5	0.55	5	5.77	5	91.46	2	3.57	2	3.27
k	82	34	776.14	50	1697.11	80	1480.55	80	1159.04	18	1275.09	37	2673.81
MutexP	4	–	–	4	27.32	4	0.26	4	27.32	–	–	4	27.32
Qshifter	4	3	38.65	4	12.07	4	4.54	4	12.07	–	–	4	12.07
s	11	11	7.01	11	1713.61	11	124.47	11	1741.41	–	–	3	35.69
SzymanskiP	9	–	–	1	83.22	2	216.69	1	83.14	–	–	1	83.23
tipdiam	30	30	2.57	30	646.91	26	765.90	30	2.57	30	1477.49	30	79.90
Total	178	98	1148.73	125	4267.31	156	3413.97	160	4022.03	62	2988.97	99	3085.87

quantor, using existential variable elimination and universal variables expansion, plus equivalence reasoning, subsumption checking, pure and unit literal detection [3].

QuBE3.0, a search-based solver with learning [9].

QuBE6.1, a composition of the search-based solver QuBE and a preprocessor that applies equivalence substitution, Q-resolution and clause subsumption.

sKizzo, a reasoning engine for QBF featuring several techniques, including search, resolution and skolemization [2].[1]

yQuaffle, a search-based solver featuring learning and inversion of quantifiers.

Table 3 shows the results of the above experiment. Table 3 is organized as Table 2, with the difference that, for each group, we added columns "#′" and "Time′", denoting the number of solved formulas (after preprocessing) and the CPU time spent to solve them (including the time of preprocessing), respectively. Moreover, for the sake of compactness, the table is split in two parts. Looking

[1] sKizzo is run with its default settings.

at Table 3, we can see that all the solvers benefit quite uniformly from QuBIS. Even more, some solvers that were not able to solve any formula for a given family, are now able to solve all these formulas. This is, e.g., the case of families `Chain` and `Qshifter` for solvers QuBE3.0 and yQuaffle and family `MutexP` for both versions of QuBE and yQuaffle. There are also cases for which the number of solved formulas increases substantially after preprocessing. This is, e.g., the case of both versions of QuBE and yQuaffle on the family `Adder`, and QuBE6.1 on the family `Counter`, wherein QuBE6.1 is now able to solve all instances.

Still with reference to Table 3, and looking at the data related to the family `k`, we can see that all solvers benefit from preprocessing. Considering the increment of solved formulas, we range from about 10% in the case of QMRes and quantor, up to 100% in the case of QuBE3.0 and yQuaffle. The only exception is sKizzo, which already performed fairly well on these instances. (80 out of 83 without preprocessing). There is also one case in which using QuBIS actually made the performances of solvers worse. This is the case of QuBE3.0 and sKizzo on the `SzymanskiP` family, wherein they solve only 1 formula after preprocessing, while they can solve 9 and 2 formulas, respectively, before preprocessing. As we said before, while the setting QuBIS$_p$ succeeds in making the formulas easier to deal with in most cases, this is not guaranteed for any setting (and in the case of QuBIS$_s$ it will most often fail to be case).

Considering the results above, it is important to notice that the impact of preprocessing on the total runtime is negligible in many cases. For instance, considering the family `Counter` the time spent by QuBIS is under 0.1 CPU seconds. The cases in which QuBIS runtime is not negligible per se, are also those where solvers get more advantage from preprocessing. For instance, looking at the performances of sKizzo on the family `k`, we can see that the time spent for solving preprocessed formulas is 78% of the time spent to solve the non-preprocessed ones, only 12% of which spent by QuBIS. However there are also cases in which time spent by QuBIS is much higher than the time used to solve the original intances. Particularly, in the case of QMRes, quantor, QuBE6.1, and sKizzo on the family `s`, the preprocessing time is about 90% of the total runtime.

Looking at the row "Total" in Table 3, we can see that, overall, all the solvers increased their performances using QuBIS. Considering only the percentage of increment in the number of solved formulas, we can see that it ranges from 2% in the case of sKizzo, up to 60% in the case of yQuaffle. Furthermore, preprocessing with QuBIS also decrease the differences between solver classes. Ordering the solvers by decreasing number of solved formulas, we can see that, considering non-preprocessed formulas the top solver is sKizzo, with 156 solved formulas, while the last one is yQuaffle with 62, solving only 40% of formulas solved by sKizzo. Looking now at results obtained using QuBIS, we can see that the first one is still sKizzo, which tops at 160 solved formuls, while the last one is now QuBE3.0, with 93 solved formulas, but QuBE3.0 is now able to solves 58% of the formulas solved by sKizzo.

5 Related Work and Conclusions

Our direct source of inspiration concerning QuBIS has been the algorithm BDR presented in [17] for SAT. From an implementation point of view, even as a proof-of-concept implementation, QuBIS is more advanced than BDR. From an algorithmic point of view, the main difference between BDR and QuBIS is that our solver uses dynamic, rather than static, reordering of variables. Since QuBIS uses Q-resolution to eliminate variables, it is in this aspect similar to QMRes and quantor. However, our approach differs from QMRes, because we do not use symbolic data structures, and also from quantor since we never expand universal variables. Besides, QuBIS is incomplete while both quantor and QMRes are not. While in the QBF literature there are other tools devoted to preprocessing formulas [18,4], QuBIS is the only one that is able to work both as an (in)complete solver and as a preprocessor. In particular, previous tools focus only on simplifying the input formula, using hyperbinary resolution in the case of [18], and a variety of techniques (including variable elimination) in the case of [4]. While both tools have can solve the input formula, this is a side effect and not the intended outcome of the tool.

In this paper we introduced a new incomplete solver for QBFs. We have showed how state-of-the-art QBF solvers can benefit by a preprocessing step performed by QuBIS. We have also showed that QBF solvers using diverse approaches to solve formulas can uniformly take advantage of QuBIS. Our future work will include an improvement of the implementation of our tool, and a deeper investigation about its ability to improve QBF solvers performances with respect to other preprocessors. We will also investigate whether the addition of other parameters, as in [7], can improve QuBIS performances.

References

1. Ansotegui, C., Gomes, C.P., Selman, B.: Achille's heel of QBF. In: Proc. of AAAI, pp. 275–281 (2005)
2. Benedetti, M.: sKizzo: a Suite to Evaluate and Certify QBFs. In: Nieuwenhuis, R. (ed.) CADE 2005. LNCS (LNAI), vol. 3632, pp. 369–376. Springer, Heidelberg (2005)
3. Biere, A.: Resolve and Expand. In: Hoos, H.H., Mitchell, D.G. (eds.) SAT 2004. LNCS, vol. 3542, pp. 59–70. Springer, Heidelberg (2005)
4. Bubeck, U., Büning, H.K.: Bounded universal expansion for preprocessing QBF. In: Marques-Silva, J., Sakallah, K.A. (eds.) SAT 2007. LNCS, vol. 4501, pp. 244–257. Springer, Heidelberg (2007)
5. Davis, M., Putnam, H.: A computing procedure for quantification theory. Journal of the ACM 7(3), 201–215 (1960)
6. Egly, U., Eiter, T., Tompits, H., Woltran, S.: Solving Advanced Reasoning Tasks Using Quantified Boolean Formulas. In: Seventeenth National Conference on Artificial Intelligence (AAAI 2000), pp. 417–422. The MIT Press, Cambridge (2000)
7. Van Gelder, A., Tsuji, Y.K.: Satisfiability testing with more reasoning and less guessing. Technical Report UCSC-CRL-95-34 (1995)

8. Giunchiglia, E., Narizzano, M., Tacchella, A.: Quantified Boolean Formulas satisfiability library (QBFLIB) (2001), www.qbflib.org
9. Giunchiglia, E., Narizzano, M., Tacchella, A.: Clause-Term Resolution and Learning in Quantified Boolean Logic Satisfiability. Artificial Intelligence Research 26, 371–416 (2006), http://www.jair.org/vol/vol26.html
10. Rowley, A.G.D., Gent, I.P., Hoos, H.H., Smyth, K.: Using Stochastic Local Search to Solve Quantified Boolean Formulae. In: Rossi, F. (ed.) CP 2003. LNCS, vol. 2833, pp. 348–362. Springer, Heidelberg (2003)
11. Jussila, T., Biere, A.: Compressing BMC Encodings with QBF. In: Proc. 4th Intl. Workshop on Bounded Model Checking (BMC 2006) (2006)
12. Kleine-Büning, H., Karpinski, M., Flögel, A.: Resolution for Quantified Boolean Formulas. Information and Computation 117(1), 12–18 (1995)
13. Narizzano, M., Pulina, L., Taccchella, A.: QBF solvers competitive evaluation (QBFEVAL) (2006), http://www.qbflib.org/qbfeval
14. Narizzano, M., Pulina, L., Tacchella, A.: QBF competition 2006 (qbfeval 2006), www.qbfeval.org/2006
15. Pan, G., Vardi, M.Y.: Symbolic Decision Procedures for QBF. In: Wallace, M. (ed.) CP 2004. LNCS, vol. 3258, pp. 453–467. Springer, Heidelberg (2004)
16. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)
17. Rish, I., Dechter, R.: Resolution versus search: Two strategies for sat. Journal of Automated Reasoning 24(1/2), 225–275 (2000)
18. Samulowitz, H., Davies, J., Bacchus, F.: Preprocessing QBF. In: Benhamou, F. (ed.) CP 2006. LNCS, vol. 4204, pp. 514–529. Springer, Heidelberg (2006)
19. Stockmeyer, L.J., Meyer, A.R.: Word problems requiring exponential time. In: 5th Annual ACM Symposium on the Theory of Computation, pp. 1–9 (1973)

Strength Two Covering Arrays Construction Using a SAT Representation

Daniel Lopez-Escogido, Jose Torres-Jimenez,
Eduardo Rodriguez-Tello, and Nelson Rangel-Valdez

Laboratorio de Tecnologías de Información, CINVESTAV-Tamaulipas,
Carretera Nacional Cd. Victoria-Monterrey Km 6, Cd. Victoria
Tamaulipas, México, CP. 87276, Tel.: (52 834) 316 6600
{dlopeze,jtj,ertello,nrangel}@cinvestav.mx

Abstract. According to the NIST report of 2002 there is a great potential to reduce the cost, and to increase the quality of the software developed in USA through the creation of automated tools that help in the software testing process. One alternative to improve the software testing process is the creation of tools that generate testing cases in an automatic way. Through the construction of Covering Arrays (CA) it is possible to obtain a minimum set of test cases with the maximum possibility of testing all the functionality of the developed software.

In this paper an approach to construct CA using the propositional satisfiability problem (SAT) is presented. The approach starts with the transformation of a CA instance into a non-conjunctive normal form (non-CNF) SAT instance. Then the SAT instance is solved through a non-CNF SAT solver, and finally the SAT solution is transformed into a CA solution. The main contributions of this work are: an efficient non-CNF SAT solver able to equals or improves previously reported results, and a simplified SAT representation to solve the CA problem.

1 Introduction

In the software industry the development and production of high-quality software at reasonable cost is a critical issue. A common source of faults in software components is the unexpected interaction between their input parameters. Software testing is used widely to assure the quality of the software that will be offered to the market. The National Institute for Standards and Technology (NIST) [19] reports that software defects cost to the USA economy close to $60 billion per year. They suggest that approximately $22 billion can be saved through a more effective software testing process. Indeed, there is a need for advanced software testing techniques that offer a good cost-benefit ratio.

Software testing process can be divided into different phases [7]: testing small pieces of code written by one programmer, integration and testing of several subsystems, and system testing (the testing of the combinations of subsystems). There are methods to test software such as structural testing or white box testing [14], referred as the type of testing in which the test cases are designed according

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 44–53, 2008.

to the architectural knowledge of the software under test. Another method is the functional testing or black box testing [14], which refers to the type of testing where only the functional knowledge of the software is known.

When a software component contains many input parameters, the risk of faults is magnified. A software development enterprise would want to exercise all the possible combinations of input parameters in each software component. However, the number of possible input parameters configurations can grow exponentially, so it becomes impractical for the enterprise to accomplish it due to its constraints of time and money. Reducing the number of test cases could produce significant savings, but it is necessary to guarantee some level of quality of the tests.

The Covering Arrays (CA) are combinatorial objects, that recently have been applied to do functional tests to software components. The use of CA allows to test all the interactions, from a given size, among input parameters using the minimum number of test cases.

According to Burr and Young [1], 80% to 90% of the functionality of a software component is exercised if the tests cover all the possible combinations of its inputs parameters in pairs. If all two-way interactions among input parameters in a software component are tested, the set of tests could be reduced. This paper introduces an approach to construct Covering Arrays using the propositional satisfiability problem (SAT).

The rest of this paper is organized as follows. Section 2 presents the CA construction problem. Section 3 gives a formal definition of SAT problem, while Section 4 introduces a model to transform a CA construction problem instance into a SAT instance. The implementation details of a Simulating Annealing algorithm used to solve the resulting SAT instances are presented in Section 5. The experiments carried out with this approach are detailed in Section 6. Finnally, the conclusions of this research work are preented in Section 7.

2 Covering Arrays

The CA have been used to test systems that involve several input parameters, such as software components, networks, and circuit systems [4,5,22,24,25]. They have been also used to elaborate experimental designs in areas like agriculture, medicine and manufacturing.

A CA with k columns, with v levels or values for each column, with an interaction size t, and N rows is defined using the following notation: $CA(N; k, v, t)$. A CA is a matrix M of size $N \times k$, where each element $m_{i,j}$ can take only one value from the set $S = \{0, 1, 2, ..., v-1\}$. The parameter v is called the alphabet of the CA and t is called the interaction size or strength. The parameter N is called the size of the CA. Given t, k and v the Covering Array Number, denoted by $N = CAN(k, v, t)$, is the minimum N for which there exists a $CA(N; k, v, t)$. A CA of size $N = CAN(k, v, t)$ is called optimal. The main goal in CA construction is to minimize the N value given k, v and t. A CA of strength two has the following property: every subset of two columns j, l $(0 \leq j < l < k)$ contains all

the elements of $\{0, 1, 2, ..., v-1\}^2$ at least once. This paper deals with the CA of strength two.

There are several approaches to construct CA; In [23] the author reported a greedy algorithm, the CA is constructed adding new rows to the matrix M one at a time according to a special heuristic. Chateauneuf and Kreher [3] use algebraic techniques for their construction. Hartmann and Raskin [12] describe a software package called Combinatorial Test Services (CTS), which finds CA with the smallest number N using a variety of constructive methods and choosing the best result. Nurmela [18] developed a Tabu Search algorithm to find CA which improved some previously reported upper bounds. The problem of finding optimal CA has solution for only some special cases, for example when the alphabet is binary and the strength is two [16], but in general, the problem is NP-complete [17].

An example of an optimal CA with 4 columns ($k = 4$), 3 possible values ($v = 3$), and strength $t = 2$, is shown in Table 1.

Table 1. A CA(9;4,3,2). Every subset of two columns j, l $(0 \leq j < l < k)$ contains all the elements of $\{0, 1, 2\}^2$.

0	0	0	0
1	0	1	1
2	0	2	2
0	2	2	1
2	1	0	1
0	1	1	2
2	2	1	0
1	1	2	0
1	2	0	2

In the following section the propositional satisfiability problem that will be used to model the CA instances is described.

3 The Propositional Satisfiability Problem

The propositional satisfiability problem has many applications in the areas of computer science, artificial intelligence, hardware design, automatic electronic design and verification, among others. The SAT problem consists in finding a truth assignment to a set of boolean variables of a formula F such that the formula F is *true* (*i.e.* the formula F is satisfied).

The SAT problem is an NP-complete problem [6]. The SAT problem involves finding an assignment to a set of boolean variables $x_i, x_2, ..., x_n$ that satisfies a set of constraints represented by a well-formed boolean formula in CNF format $F : B^n \rightarrow B$, $B = \{0, 1\}$, *i.e.*, a conjunction of clauses, each of which is a disjunction of variables. If such assignment exists the instance is called $satisfiable$ and $unsatisfiable$ otherwise.

The SAT problem has been used as a language to encode several kinds of hard problems, and in many cases solving a problem in the SAT domain is easier than solving the original problem. In this work SAT is used to model CA construction problem instances. The following section presents this transformation model.

4 Transformation Model

The approach presented in this paper produces a non-CNF formula to model instances of CA of strength two. The model for a $CA(N; k, v, 2)$ instance requires for each element $m_{i,j}$ of the matrix M associated with the CA instance the use of v boolean variables $m_{i,j,x}$ $(0 \leq x < v)$. The element $m_{i,j}$ of the matrix M takes the value x iff the boolean variable $m_{i,j,x}$ is *true*. The clauses required are presented in the Equations 1, 2 and 3.

$$\kappa_1 = \bigwedge_{i=0}^{N-1} \Big(\bigvee_{\forall j,x | 0 \leq j < k, 0 \leq x < v} m_{i,j,x} \Big) \tag{1}$$

$$\kappa_2 = \bigwedge_{i=0}^{N-1} \Big(\bigvee_{\forall j,x,y | 0 \leq j < k, 0 \leq x < y < v} (\overline{m_{i,j,x}} \vee \overline{m_{i,j,y}}) \Big) \tag{2}$$

$$\kappa_3 = \bigwedge_{\forall x,y | 0 \leq x \leq y < v} \Big(\bigwedge_{\forall j,l | 0 \leq j < l < k} \Big(\bigvee_{\forall i | 0 \leq i < N} (m_{i,j,x} \wedge m_{i,l,y}) \Big) \Big) \tag{3}$$

The Equation 1 defines clauses that guarantee that each element in the CA matrix M takes at least a value from $\{0, 1, ..., v-1\}$; it generates $N \times k$ clauses, and the size of every clause is v. The clauses defined by the Equation 2 specify that each element in the CA matrix M takes only one value; the total number of clauses produced by this equation are $N \times k \times \binom{v}{2}$, each one of size 2. The clauses shown in Equations 1 and 2 are in CNF.

The third set of clauses, shown in the Equation 3, guarantee that the CA is constructed correctly according to its definition, *i.e.* that the interactions among the columns is met. These clauses are organized in a non-CNF. The SAT formula that represents a CA instance will include the three set of clauses previously described, as shown in the Equation 4.

$$F = \kappa_1 \wedge \kappa_2 \wedge \kappa_3 \tag{4}$$

The SAT model for CA(4;2,2,2) is given in Equations 5, 6 and 7.

$$\begin{aligned} \kappa_1 = {} & (m_{0,0,0} \vee m_{0,0,1}) \wedge (m_{0,1,0} \vee m_{0,1,1}) \wedge \\ & (m_{1,0,0} \vee m_{1,0,1}) \wedge (m_{1,1,0} \vee m_{1,1,1}) \wedge \\ & (m_{2,0,0} \vee m_{2,0,1}) \wedge (m_{2,1,0} \vee m_{2,1,1}) \wedge \\ & (m_{3,0,0} \vee m_{3,0,1}) \wedge (m_{3,1,0} \vee m_{3,1,1}) \end{aligned} \tag{5}$$

$$\begin{aligned} \kappa_2 = {} & (\overline{m_{0,0,0}} \vee \overline{m_{0,0,1}}) \wedge (\overline{m_{0,1,0}} \vee \overline{m_{0,1,1}}) \wedge \\ & (\overline{m_{1,0,0}} \vee \overline{m_{1,0,1}}) \wedge (\overline{m_{1,1,0}} \vee \overline{m_{1,1,1}}) \wedge \\ & (\overline{m_{2,0,0}} \vee \overline{m_{2,0,1}}) \wedge (\overline{m_{2,1,0}} \vee \overline{m_{2,1,1}}) \wedge \\ & (\overline{m_{3,0,0}} \vee \overline{m_{3,0,1}}) \wedge (\overline{m_{3,1,0}} \vee \overline{m_{3,1,1}}) \end{aligned} \tag{6}$$

$$\begin{aligned}\kappa_3 = &((m_{0,0,0} \wedge m_{0,1,0}) \vee (m_{1,0,0} \wedge m_{1,1,0}) \vee \\ &(m_{2,0,0} \wedge m_{2,1,0}) \vee (m_{3,0,0} \wedge m_{3,1,0})) \wedge \\ &((m_{0,0,0} \wedge m_{0,1,1}) \vee (m_{1,0,0} \wedge m_{1,1,1}) \vee \\ &(m_{2,0,0} \wedge m_{2,1,1}) \vee (m_{3,0,0} \wedge m_{3,1,1})) \wedge \\ &((m_{0,0,1} \wedge m_{0,1,0}) \vee (m_{1,0,1} \wedge m_{1,1,0}) \vee \\ &(m_{2,0,1} \wedge m_{2,1,0}) \vee (m_{3,0,1} \wedge m_{3,1,0})) \wedge \\ &((m_{0,0,1} \wedge m_{0,1,1}) \vee (m_{1,0,1} \wedge m_{1,1,1}) \vee \\ &(m_{2,0,1} \wedge m_{2,1,1}) \vee (m_{3,0,1} \wedge m_{3,1,1}))\end{aligned} \tag{7}$$

In the literature Hnich *et al.* [13] reported a model that transforms CA instances into SAT instances. The model reported in [13] uses $N \times k \times v + N \times \binom{k}{2} \times v^2$ boolean variables and $2N \times k \times v + v^2 \times N \times \binom{k}{2} + 2N \times v^2 \times \binom{k}{2}$ literals. The non-CNF model described in this paper only requires $N \times k \times v$ boolean variables, and $2N \times (k + k \times \binom{v}{2} + v^2 \times \binom{k}{2})$ literals. If both models are compared then it will be evident that the search space in the model reported in [13] is greater than that produced by the model proposed here.

On the other hand, no model that uses $N \times k \times v$ boolean variables to transform CA into a CNF SAT instance was found in the literature. An alternative is to apply the distributive law to Equation 3 but it will result in an exponential growth in the number of literals. In particular, the number of literals produced will be $v^2 \times \binom{k}{2} \times 2^N$.

In order to solve the non-CNF SAT model, a Simulated Annealing metaheuristic was used, implementation details are described in the next section.

5 A Simulated Annealing Implementation

The SAT problem is an issue widely studied, and many researchers have developed many approaches to solve different instances of this problem. There are two main approaches to construct solvers for SAT instances, they are complete SAT solvers and incomplete SAT solvers. The complete SAT solvers usually are based on the Davis-Putnam procedure [8,9], which employs a systematic backtracking search to explore the solution space, looking for a satisfying assignment. The main inconvenient with this procedure is that the execution time can grow exponentially according to the problem size (the number of boolean variables).

The incomplete SAT solvers employ mainly local search procedures, these solvers may find a solution in less time than the complete SAT solvers, but there is not guarantee that a solution can be found even if it exists. Local search algorithms have shown to be very effective for solving SAT instances [21], specially when the size of the instance does not allow to use an exact algorithm [10,11,20].

In this work a local search implementation based on a classic Simulated Annealing (SA) [2,15] was used to construct an incomplete SAT solver for the proposed model. This solver avoids the use of the set of clauses shown in Equations 1 and 2 by constructing a random initial solution, that ensures the correct assignment of values to elements of the matrix M. The correct assignment of values implies that each element $m_{i,j}$ of the matrix M will take only one value.

It is formally defined as follows: let $m_{i,j,x}$ be the set of boolean variables created by model from a CA instance then if $m_{i,j,x} \leftarrow true$ then $m_{i,j,y} \leftarrow false, \forall x \neq y$.

The evaluation function for the SA returns the number of unsatisfied clauses. In order to find a solution for the CA instance the number of unsatisfied clauses must be zero. The solution will be a matrix M where the value of each element $m_{i,j}$ will be x iff the boolean variable $m_{i,j,x} = true$.

In the algorithm a set of p solutions is created from the current solution by randomly selecting two variables from it and exchanging their values. The neighboring function then chose as the new solution the neighbor, the solution with the best value of the evaluation function.

After that, the algorithm will form a new neighbor by exchanging values between respect to the matrix M.

In the next section the experiments carried out with this SA algorithm are presented.

6 Experiments and Results

The model introduced in Section 4 was used to model different strength two CA instances, and for solving them. This section presents the main results obtained. In a preliminary experiment a complete SAT solver for this representation was implemented. It only used the set of clauses 3; and it employed the Davis-Putnam procedure [8,9]. The complete solver showed good performance in small cases (see Table 2), but in CA instances with more than four columns and alphabets greater than five the execution time grew up exponentially.

Due to the excessive consumed time consumed by this complete SAT solver, a second experiment was designed. This experiment used an incomplete SAT solver based on the SA approach described in Section 5. The parameters used for the SA were: an initial temperature $c_0 = 1$, a final temperature $c_f = 1 \times 10^{-9}$, a cooling factor $\alpha = 0.99$, and a Markov chain length $L = N \times k \times v$. The number of solutions used for the neighboring function is $p = 20 \times v$.

Since the optimum value of N is unknown we defined a lower bound and an upper bound, then a binary search technique was used to calculate a new value for N. The process is repeated with this new value of N. This procedure ends when it is not possible to lower the value of N anymore. Elementary counting arguments show that the minimal size of a CA of strength two is v^2, then v^2 was used as the lower bound. As the upper bound we selected p^2, were p is the

Table 2. Strength two Covering Arrays constructed using a complete SAT solver

k	v	N	**time[sec]**
3	3	9	0.04
4	3	9	0.04
3	4	16	0.47
4	4	16	1.73
5	4	16	3.61
3	5	25	518.38

Table 3. Results for $CA(k, v, 2)$. This table shows the results obtained by the non-CNF solver implemented in this work. Its results are compared with those reported in the literature. HW, HR, CK, NU denotes [13,12,3,18] respectively (results in italics are not the optimal value, these are the best upper bounds reported).

		N						
k	v	**non-CNF**	**HW**	**HR**	**CK**	**NU**	Optimal	time[sec]
3	3	9	9	9	9	-	9	0.002
3	4	16	16	16	16	-	16	0.027
3	5	25	25	25	25	-	25	0.53
3	6	36	36	36	36	-	36	1.1
3	7	49	49	49	49	-	49	6.93
4	3	9	9	9	9	-	9	0.007
4	4	16	16	16	16	-	16	0.034
4	5	25	25	25	25	-	25	0.95
4	6	37	37	48	37	37	*37*	32.1
4	7	49	49	49	49	-	49	3589.2
5	3	11	11	15	11	11	*11*	0.6
5	4	16	16	16	16	-	16	1.3
5	5	25	25	25	25	-	25	5.27
5	6	39	39	48	39	39	*39*	5621.42
5	7	52	52	49	49	-	49	7237.39
6	3	12	12	15	12	-	*12*	0.99
6	4	19	19	24	19	19	*19*	338.24
6	5	25	25	25	25	-	25	2360.52
6	6	42	42	48	41	41	*41*	4070.16
6	7	**57**	58	49	49	-	49	4622.93
7	3	12	12	15	12	-	*12*	121.83
7	4	21	21	28	21	21	*21*	782.12
7	5	29	29	45	29	29	*29*	3125.23
7	6	**44**	45	48	42	-	*42*	3223.93
7	7	**60**	61	49	49	-	49	7085.96
8	3	14	14	15	13	-	*13*	0.63
8	4	23	23	28	23	23	*23*	10.23
8	5	34	34	45	33	33	*33*	6552.51
8	6	**47**	48	48	42	42	*42*	6842.18
8	7	63	63	49	49	-	49	7152.83

smallest prime greater than k and v, the logic behind this is that it is possible to construct a $CA(p^2; k, v, 2)$ by using a $CA(p^2, p+1, p, 2)$ [12].

Both, the complete and incomplete SAT solvers were coded in C and compiled with gcc using -O3 and -lm as parameters for the compiler. They were run into a Pentium 4 at 2.8 GHz with 1 GB of RAM under a Linux operating system.

The incomplete SAT solver was compared with four approaches reported in the literature: Nurmela (NU) solved CA using tabu search [18], Chateauneuf and Kreher (CK) used mathematical constructions [3], Hartman and Raskin (HR)

[12] used direct and recursive methods and Hnich *et al.* [13] (HW) using a CNF SAT solver.

The results obtained in the experimentation are shown in Table 3. The columns 1 and 2 indicate the number of columns and the CA alphabet for each instance. Column 3 shows the results obtained using the approach presented in this paper. When the font is bold indicates that the results obtained by the approach presented in this paper is better than the reported in [13]. The columns 4, 5, 6 and 7 present the results reported by [13,12,3,18], respectively. Column 8 depicts the best reported results for each instance; when the number appears in italics corresponds to the best known solution, otherwise it is an optimal value. The column 9 shows the CPU time in seconds spent by our approach to solve each instance.

7 Conclusions

In this work a model to transform CA of strength two into the SAT problem instances was presented. The SAT instance produced is in a non-CNF form and is smaller, in the number of literals and boolean variables, than its equivalent in CNF form previously reported in [13].

An incomplete SAT solver based on SA was implemented. This solver used local search to construct the CA for a benchmark composed of 30 instances.

It was experimentally shown that it is possible to construct in a competitive way CA using a non-CNF SAT formula that represents the problem. Furthermore, it was shown that the performance of the incomplete SAT solver is good for reasonable size CA instances.

The approach presented in this paper enable us to equal and even improve some results produced by a previously similar approach reported.

Acknowledgments

This research was partially funded by the following projects: CONACyT 58554 Cálculo de Covering Arrays, 51623 Fondo Mixto CONACyT y Gobierno del Estado de Tamaulipas, CONACyT 74521 Repatriación.

References

1. Burr, K., Young, W.: Combinatorial test techniques: Table-based automation, test generation, and code coverage. In: Proceedings of the International Conference on Software Testing Analysis and Review, San Diego, USA, pp. 503–513 (October 1998)
2. Cerny, V.: A thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal Optimization Theory and Applications 45, 41–51 (1985)

3. Chateauneuf, M., Kreher, D.L.: On the state of strength-three covering arrays. Journal of Combinatorial Designs 10(4), 217–238 (2002)
4. Cohen, D.M., Parelius, J., Dalal, S.R., Patton, G.C.: The combinatorial design approach to automatic test generation. IEEE Sofware 13(5), 83–88 (1996)
5. Cohen, D.M., Dalal, S.R., Fredman, M.L., Patton, G.C.: The aetg system: an approach to testing based on combinatorial design. Transactions on Software Engineering 23(7), 437–444 (1997)
6. Cook, S.A.: The complexity of theorem-proving procedures. In: STOC 1971: Proceedings of the third annual ACM symposium on Theory of computing, pp. 151–158. ACM, New York (1971)
7. Dalal, S.R., Mallows, C.L.: Factor-covering designs for testing software. Technometrics 40(3), 234–243 (1998)
8. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. Communications. ACM 5(7), 394–397 (1962)
9. Davis, M., Putnam, H.: A computing procedure for quantification theory. J. ACM 7(3), 201–215 (1960)
10. Gu, J.: Efficient local search for very large-scale satisfiability problems. SIGART Bull. 3(1), 8–12 (1992)
11. Hansen, P., Jaumard, B.: Algorithms for the maximum satisfiability problem. Computing 44(4), 279–303 (1990)
12. Hartman, A., Raskin, L.: Problems and algorithms for covering arrays. Discrete Math. 284(1-3), 149–156 (2004)
13. Hnich, B., Prestwich, S.D., Selensky, E., Smith, B.M.: Constraint models for the covering test problem. Constraints 11(2-3), 199–219 (2006)
14. Jorgensen, P.C.: Software Testing: A Craftsman's Approach. CRC Press, New York (2002)
15. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 4598, 671–680 (1983)
16. Kleitmain, D.J., Spencer, J.: Families of k-independent sets. Discrete Math. 6, 255–262 (1973)
17. Lei, Y., Tai, K.C.: In-parameter-order: A test generation strategy for pairwise testing. In: HASE 1998: The 3rd IEEE International Symposium on High-Assurance Systems Engineering, Washington, DC, USA, pp. 254–261. IEEE Computer Society, Los Alamitos (1998)
18. Nurmela, K.J.: Upper bounds for covering arrays by tabu search. Discrete Applied Math. 138(1-2), 143–152 (2004)
19. National Institute of Standards and Technology. The economic impacts of inadequate infrastructure for software testing (2002), www.nist.gov/director/prog-ofc/report02-3.pdf
20. Selman, B., Kautz, H.A., Cohen, B.: Local search strategies for satisfiability testing. In: Proceedings of the Second DIMACS Challange on Cliques, Coloring, and Satisfiability (1993)
21. Selman, B., Levesque, H., Mitchell, D.: A new method for solving hard satisifiability problems. In: AAAI 1992, San Jose, CA, pp. 440–446 (July 1992)
22. Sloane, N.J.A.: Covering arrays and intersecting codes. Journal of Combinatorial Designs 1(1), 51–63 (1993)
23. Turban, R.C.: Algorithms for covering arrays. PhD thesis, Tempe, AZ, USA, Adviser-Charles Colbourn (2006)

24. Williams, A.W., Probert, R.L.: A practical strategy for testing pair-wise coverage of network interfaces. In: ISSRE 1996: Proceedings of the Seventh International Symposium on Software Reliability Engineering (ISSRE 1996), Washington, DC, USA, pp. 246–256. IEEE Computer Society, Los Alamitos (1996)
25. Williams, A.W.: Determination of test configurations for pair-wise interaction coverage. In: TestCom 2000: Proceedings of the IFIP TC6/WG6.1 13th International Conference on Testing Communicating Systems, Deventer, The Netherlands, pp. 59–74. Kluwer, B.V. (2000)

A New Framework for Local Belief Revision

Omar Doukari, Robert Jeansoulin, and Eric Würbel

Laboratoire LSIS, UMR CNRS 6168
Domaine Universitaire de Saint-Jérôme
13397 MARSEILLE CEDEX 20 - France

Abstract. AGM-style revision consists of two suboperations, contraction followed by expansion. With respect to Hansson, this is called "internal revision" and an alternative procedure, "external revision" is proposed. It consists in, first, expanding the belief base by the new sentence and after that contracting by its negation. In this paper, on the one hand, we propose a new framework for local belief revision by extending the LS-model introduced by Parikh and studied by several authors in the last decade. The new model, called the C-structure model, is based on adapting the containment property in propositional logic, since it has been defined in a spatial context. On the other hand, we define a "local external revision" operation equivalent to the one defined by Hansson -in a classical (not-local) framework- by defining local contraction operation.

1 Introduction

In knowledge representation, a rational agent tries to translate a mix of perceptions and beliefs. As agents often face incomplete, uncertain, and inaccurate information, they need a revision operation in order to manage possible belief changes in presence of new information. The agent's epistemic state represents its reasoning process; belief revision consists of modifying its initial epistemic state in order to maintain consistency, while keeping new information and modifying the least possible previous information.

In belief revision, much work takes as its starting point the work of Alchourrón, Gärdenfors and Makinson [1] who proposed and investigated a set of postulates, widely known as the AGM postulates, which appear to capture much of what characterizes rational belief revision. In their framework (developed for belief sets), the revision by α consists of two suboperations, contraction by $\neg\alpha$ takes place first and is then followed by expansion by α. With respect to Hansson [2], this is called "internal revision" and an alternative procedure, "external revision" is proposed. It consists in, first, expanding the belief base by α and after that contracting by $\neg\alpha$.

It turns out that in the general case the theoretical complexity of revision is high. More precisely, it belongs to the $\prod_2^p$ class in the framework of propositional logic [3,4]. The same problem for the few applications which have been developed for belief revision is posed [5].

In this paper, on the one hand, we propose a new framework for local belief revision by extending the LS-model (Language Splitting Model) defined, firstly,

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 54–64, 2008.

by Parikh [6] and studied by several authors in the last decade [7,8,9,10,11]. The new model, called the C-structure model, is based on adapting containment property in propositional logic since it has been defined in a spatial context [12]. This property assumes the limitation of the maximal size of eventual existing minimal inconsistencies.

on the other hand, since the revision problem is known to be a difficult problem, and there does not exist any revision approach, really effecient, that can treat real applications with a huge amount of data [5,13,14,4,3], we define a "local external revision" operation equivalent to the one defined by Hansson -in a classical (not-local) framework- by defining local contraction operation.

After giving some definition on kernel contraction, in Section 2, we devote Section 3 to the presentation of the LS-model. In Section 4 we define the C-structure model. In Section 5 and 6, we present our local revision operator and some complexity properties of this operator before concluding in Section 7.

Notation: In the following, $\mathcal{L}$ is a propositional language defined on a finite set of propositional variables (atoms) $\mathcal{V}$ and the usual connectors ($\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$). If α is a sentence then $\mathcal{V}(\alpha)$ denotes the set of atoms composing the sentence α. If V is a subset of $\mathcal{V}$ then $\mathcal{L}(V)$ represents the propositional sublanguage defined over V. If X is a set of sentences then $Cn(X)$ is the logical closure of X. In particular, a subset T of $\mathcal{L}$ is a belief set (theory) iff $T = Cn(T)$. We shall use letters T, T' etc. for theories, and B_T is a belief base of the belief set T iff B_T is a finite subset of T and $Cn(B_T) = T$. In particular, if B_T is an inconsistent belief base, we say that $M \subseteq B_T$ is a MIS of B_T i.e., a minimal inconsistent subset of sentences of B_T iff M is inconsistent and for all $M' \subset M$, M' is consistent. In the following, B_T denotes an arbitrary belief base of T. $T * \alpha$ is the revision of T by α, $T - \alpha$ is the contraction of T by α and finally, $T + \alpha$ is the operation of expansion of T by α, it is equal to $Cn(T \cup \{\alpha\})$, i.e. the result of a brute addition of α to T (followed by logical closure) without considering the need for consistency. Constructing $T * \alpha$ is contracting T by $\neg\alpha$ then adding α. Formally, this construction is given by the LEVI identity [2]: $T * \alpha = (T - \neg\alpha) + \alpha$.

2 Kernel Contraction

The contraction of a set of sentences A by α according to kernel contraction introduced by Hansson in [15] consists in selecting among the sentences of a set A that contribute *effectively* to imply α; then to remove at least one element of each selected subset (i.e. a Hitting set of the selected subsets) so that the result does not imply α.

The subsets of a set A that contribute effectively to imply α are the minimal subsets of A implying α. Formally:

Definition 1. *[15] Let A be a set in $\mathcal{L}$ and α a sentence. Then $A\perp\alpha$ is a set such that $B \in A\perp\alpha$ iff:* (i) $B \subseteq A$, (ii) $\alpha \in Cn(B)$, *and* (iii) *If $B' \subset B$ then $\alpha \notin Cn(B')$.*

$A\perp\alpha$ is called the kernel set of A wrt α and its elements are the α-kernels of A.

To calculate a hitting set of the kernel set of A we define an incision function which is a function defined from sets of sets of sentences into sets of sentences, selecting at least one sentence from each set of the argument. Formally:

Definition 2. *[15] Let A be a set of sentences. An incision function σ is a function such that for all sentences α we have:* (i) $\sigma(A\bot\alpha) \subseteq \bigcup(A\bot\alpha)$, *and* (ii) $\forall B \in A\bot\alpha$, *and* $B \neq \emptyset$ *then* $B \cap \sigma(A\bot\alpha) \neq \emptyset$.

Thus, kernel contraction is defined as follows:

Definition 3. *[15] Let A be a set of sentences and σ an incision function for A. Kernel contraction $-_\sigma$ for A is defined as follows:* $A -_\sigma \alpha = A \setminus \sigma(A\bot\alpha)$.

An operator $-$ for a set A is a kernel contraction iff there is an incision function σ for A such that $A - \alpha = A -_\sigma \alpha$ for all sentences α.

Hansson also provided an axiomatic characterisation for kernel contraction.

Theorem 1. *[15] The operator $-$ for a set of sentences A is a kernel contraction iff for all sentences α, it satisfies:*

- *Success : If $\alpha \notin Cn(\emptyset)$, then $\alpha \notin Cn(A - \alpha)$.*
- *Inclusion : $A - \alpha \subseteq A$.*
- *Uniformity : If $\forall A' \subseteq A, \alpha \in Cn(A')$ iff $\beta \in Cn(A')$ then $A - \alpha = A - \beta$.*
- *Core-retainment : If $\beta \in A$ and $\beta \notin A - \alpha$ then there is some set A' such that $A' \subseteq A$ and $\alpha \notin Cn(A')$ but $\alpha \in Cn(A' \cup \{\beta\})$.*

3 The Language Splitting Model

The intuition behind the language splitting model is that our beliefs are subdivided into *disjoint* areas which do not affect each other [6].

Definition 4. *Let T be a theory of $\mathcal{L}$ and let $V_1, ..., V_n$ be a partition of $\mathcal{V}$. Then $\{V_1, ..., V_n\}$ is a T-splitting if $\forall i \in \{1 \dots n\}$ there exist sentences $\alpha_i \in \mathcal{L}(V_i)$, s.t., $T = Cn(\bigcup_{i=1}^{n}\{\alpha_i\})$.*

The disjoint sublanguages assumption of the LS-model allows us to revise our beliefs *locally* and to minimize the amount of computation we have to do when we revise a new piece of information, both in checking whether it is consistent with the old set of beliefs and also in revising our beliefs in view of the new information. For example, an agent that is revising his beliefs about planetary motion is unlikely to revise his beliefs about Malaysian politics. This simple intuition is not fully captured in the AGM paradigm.

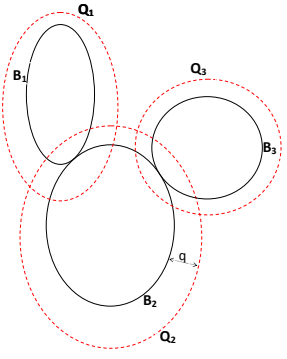
In [6] Parikh introduced a new axiom (P), as a supplement to the AGM postulates, which is defined as follows.

(P) : If $T = Cn(\mathcal{X}, \mathcal{Y})$ where $\mathcal{X}, \mathcal{Y}$ are sentences of disjoint sublanguages $\mathcal{L}_1, \mathcal{L}_2$ respectively, and $\alpha \in \mathcal{L}_1$, then $T * \alpha = (Cn_{\mathcal{L}_1}(\mathcal{X}) \circ \alpha) + \mathcal{Y}$, where $\circ$ is a revision operator of the sublanguage $\mathcal{L}_1$.

The language splitting model requires an agent's beliefs be partitioned into theories which have totally disjoint languages. In practice, however, beliefs in different areas do have some overlap in subject matter and so the partition of the main language is not actually strict. An agent's component theories do contain beliefs that are more relevant to one another than to beliefs in other component theories, but they are not totally irrelevant to beliefs in other subtheories.

In the following section, we extend the language splitting model by defining the C-structure model. This model allows some overlap between sublanguages using containment property defined in [12], which has been proposed in order to revise spatial information by splitting up space into different subspaces and revising each one separately. The idea of this property is:"*if a restriction of the belief base to a subspace is consistent, and if such a restriction is consistent with the belief base attached to the q-covering*[1] *of this subspace, then this restriction is consistent with any other information*".

Example 1. we consider a geographical space of two dimensions, subdivided into subspaces B_1, B_2 and B_3. We denote by $SB(X)$ the subset of sentences attached to subspace X. Q_1, Q_2 and Q_3 are the q-coverings of B_1, B_2 and B_3 respectively. $SB(B_3)$ is outside the q-covering of B_1, so there exists some independence between their minimal inconsistent subsets of sentences (MISs). If $SB(B_1 \cup Q_1)$, $SB(B_2)$ and $SB(B_3)$ are consistent, then $SB(B_1)$ is consistent with the subset of sentences $SB(B_2)$ and $SB(B_3)$.



So containment property is based on the restriction of the *size* of existing MISs by q, the *thickness* of q-coverings. Thus by generalizing containment property which was defined in a spatial context by defining it according to the k-relevance relation between atoms, we extend the language splitting model to become the C-structure model.

4 The *C*-Structure Model

Usually inconsistency is due to the accidental presence of a "few" pieces of contradictory information about a given subject, moreover large globally inconsistent problems are, usually, generated artificially since they are scarce in real life applications [16] (see e.g. the flooded valley application [5], the pigeons-holes problem [17], Tseitin and Urqhart's formulas [3], etc).

The language splitting model is based on properties of the agent which carries out the revision like principle of minimal change, limited capacities of real agents, etc, and on properties of belief sets (or belief bases) like modularity. However, it

[1] A q-covering of a subspace B is the subspace Q covering B for some distance which is equal to q. For further detail, see [12].

does not take into account the properties of the inconsistencies which may exist in real belief bases like those of the field of spatial information. Among these properties we find containment property which assumes that real life applications are locally inconsistent, that means the maximal size of existing MISs is limited by a certain "distance" depending on the application traited.

For the same motivations as those given in the language splitting model, namely minimal change, and by taking into account containment property, we define a new model called the C-structure model, which keeps the principle of language splitting model (disjoint sublanguages) to define a set of *cores* of a given language, and each core has a *covering* of atoms surrounding it. This concept of covering allows us some degree of overlapping between the different sublanguages.

Now, using these two concepts (core and covering) we extend the previous model while avoiding the hard assumption put for it (the assumption of the disjoint sublanguages). Our working hypothesis is a weak one with respect to that of the language splitting model. It is related to the maximal "size" of eventual existing MISs.

We begin by defining a set of cores of $\mathcal{L}$ as a partition of a set $\mathcal{V}$.

Definition 5. *$\{V_1, ..., V_n\}$ is a set of cores of $\mathcal{L}$ iff it is a partition of $\mathcal{V}$.*

Example 2. Let $\mathcal{L}$ be the propositional language defined over $\mathcal{V} = \{a, b, c, d, e, f, g, h, i, j, k, l\}$. Let T be an arbitrary theory defined on $\mathcal{L}$ and axiomatized by B_T the following belief base:

$$\left\{ \begin{array}{c} a \vee b, \neg c, b \rightarrow c, \neg d, \\ c \rightarrow (d \vee e), e \leftrightarrow f, \\ f \rightarrow g, \neg g \vee h, i \rightarrow h, \\ i, j \rightarrow i, j, k \vee l \vee j \end{array} \right\}$$

The set $\{\{a, b, c\}, \{d, e, f\}, \{g, h, i\}, \{j, k, l\}\}$ is a set of cores of $\mathcal{L}$.

Now, to order the atoms of the language $\mathcal{L}$, we use the following relevance relation inspired from [18].

Definition 6. *Let B_T be a belief base of a theory T. We say that two atoms, p and q, are directly relevant wrt B_T, denoted by $R(p, q, B_T)$ (or by $R_0(p, q, B_T)$), iff:* (i) $\exists \alpha \in B_T$ *s.t., $p, q \in \mathcal{V}(\alpha)$, or* (ii) $p = q$. *Two atoms p, q are k-relevant wrt B_T, denoted by $R_k(p, q, B_T)$, if $\exists p_0, p_1, ..., p_{k+1} \in \mathcal{V}$ s.t.: $p_0 = p$; $p_{k+1} = q$; and $\forall i \in \{0, ..., k\}, R(p_i, p_{i+1}, B_T)$.*

In Example 2, we have: $R(a, b, B_T)$, $R_1(a, c, B_T)$, $R_2(a, d, B_T)$, etc.

We introduce from this definition the concept of neighborhood as the following.

Definition 7. *Let B_T be a belief base of a theory T. Two atoms $p, q \in \mathcal{V}$ are neighbors wrt B_T iff: $\exists k \geq 0$ such that $R_k(p, q, B_T)$.*

To define clearly the degree of overlapping that takes place between the various sublanguages and to quantify local inconsistencies (MISs), we need to define a distance between variables.

Definition 8. *Suppose two atoms $p, q \in \mathcal{V}$, B_T a belief base of a theory T. The distance between p, q wrt B_T, denoted by $dist(p, q, B_T)$, is defined as follows.*

$$dist(p, q, B_T) = \begin{cases} min\{k | R_k(p, q, B_T)\} & \text{if } p, q \text{ are neighbors} \\ \infty & \text{otherwise.} \end{cases}$$

In Example 2, we have: $dist(a, b, B_T) = 0$, $dist(a, c, B_T) = 1, dist(a, d, B_T) = 2$, etc.

The covering whose thickness is equal to k for a core V_i is defined as follows.

Definition 9. *Let $\{V_1, ..., V_n\}$ be a set of cores of $\mathcal{L}$ and B_T be a belief base of a theory T. We say that $Cov_k(V_i, B_T)$ is a covering whose thickness is equal to k of V_i wrt B_T iff: $Cov_k(V_i, B_T) \subseteq \mathcal{V}$; and $\forall p \in \mathcal{V}, if \exists q \in V_i$ s.t., $dist(p, q, B_T) \leq k$ then $p \in Cov_k(V_i, B_T)$.*

For example, the set of coverings, for $k = 0$, corresponding to the set of cores $\{\{a, b, c\}, \{d, e, f\}, \{g, h, i\}, \{j, k, l\}\}$ wrt B_T (Example 2) is: $\{\{a, b, c, d, e\}, \{c, d, e, f, g\}, \{f, g, h, i, j\}, \{i, j, k, l\}\}$.

Since our model is based on the size of MISs, we quantify them as follows.

Definition 10. *Let B_T be an inconsistent belief base. Let M be a MIS of B_T. The size of M, wrt B_T, denoted by $Size(M, B_T)$ is such that: $Size(M, B_T) = max\{dist(a, b, B_T) | a, b \in \mathcal{V}(M)\}$.*

For example (Example 2), if $M = \{\neg c, c \vee d, \neg d\}$ then, $Size(M, B_T) = 0$.

A C-structure is a set of structures such that each one of them is composed of two parts. The first part called the *sublanguage part* which is composed of a core and its covering, and the second one is the *subbase part* which is a belief subbase defined on the sublanguage part.

Informally, a C-structure represents a thematic view of the overall knowledge of an agent with a good understanding of the interactions between subjects.

In this paper, the only hypothesis, that we make, is that the maximal size of eventual existing MISs in a given belief base is known. Hence, when we want to construct a C-structure C of a belief base B_T, we only require that the thickness of coverings of cores (value of k) should be (at least) equal to the maximal size of MISs which may exist in B_T. More formally:

Definition 11. *Let T be a theory defined on $\mathcal{L}$ and B_T an arbitrary belief base of T. The set $\{(V_1, Cov_k(V_1, B_T), B_1), ..., (V_n, Cov_k(V_n, B_T), B_n)\}$ is a C-structure of B_T iff:* (i) *$\{V_1, ..., V_n\}$ is a set of cores of $\mathcal{L}$,* (ii) *$\{Cov_k(V_1, B_T), ..., Cov_k(V_n, B_T)\}$ is the corresponding set of coverings wrt B_T s.t., $\forall i \in \{1, ..., n\} \forall \alpha \in \mathcal{L}(Cov_k(V_i, B_T))$, if $B_T \cup \{\alpha\}$ is inconsistent, then $\forall M$ a MIS of $B_T \cup \{\alpha\}$, $Size(M, B_T) \leq k$, and* (iii) *$\forall B_i$, $B_i = \mathcal{L}(Cov_k(V_i, B_T)) \cap B_T$.*

We denote by $\bigcup_{i=1}^{n} B_i$, the informational part of the C-structure. We shall use the letter C both for a C-structure C and for the informational part of C. It will be clear from the context which is meant.

We obtain the following C-structure corresponding to Example 2 by assuming that the maximal size of eventual existing MISs in B_T is 0.
$\{(\{a, b, c\}, \{a, b, c, d, e\}, \{a \vee b, \neg c, b \rightarrow c, \neg d, c \rightarrow (d \vee e)\}), (\{d, e, f\}, \{c, d, e, f, g\},$

$\{\neg c,\ c \rightarrow (d \vee e),\ \neg d,\ e \leftrightarrow f,\ f \rightarrow g\}), (\{g,\ h,\ i\},\ \{f, g, h, i, j\},\ \{f \rightarrow g,\ \neg g \vee h,\ i \rightarrow h,\ i,\ j \rightarrow i,\ j\}), (\{j,\ k,\ l\},\ \{i, j, k, l\},\ \{i,\ j \rightarrow i,\ j,\ k \vee l \vee j\})\}$.

Hence, if all subbases part of C are consistent then C is globally consistent. Formally, we deduce the containment property as follows.

(Containment Property): If $\{(V_1, Cov_k(V_1, B_T), B_1), ..., (V_n, Cov_k(V_n, B_T), B_n)\}$ is a C-structure C of B_T, then: if $\forall i \in \{1, ..., n\}, B_i$ is consistent, then C is globally consistent.

5 *C*-Structure Revision

AGM-style revision by α consists of two suboperations, expansion by α and contraction by $\neg\alpha$. In the AGM revision operation (first developed for belief sets), contraction by $\neg\alpha$ takes place first and is then followed by expansion by α. In [2] this is called "internal revision" and an alternative procedure, "external revision" is proposed. It consists in first expanding the belief base by α and after that contracting by $\neg\alpha$.

Hence, we will assume that local external revision consists of (local) expansion followed by local contraction (local expansion in this context consists simply, in adding new sentences according to the sublanguage in which they are defined. For example, if we want to add a sentence α to a C-structure C of B_T, we must add it to all structures of C whose their cores V_i are such that $\mathcal{V}(\alpha) \subseteq Cov_k(V_i, B_T)$. So, new sentences can be added to some structures of C and not to all structures).

5.1 Local Contraction

The idea behind kernel contraction is that, if we remove from the belief base at least one element of each α-kernel, we obtain a belief base that does not imply α. In the case of the local operation, we are interested in obtaining a belief subbase defined over a sublanguage of C that does not imply α. We consider then only the kernel set belonging to this belief subbase and use an incision function to select the sentences from the kernel set to be removed.

Definition 12. *Let* $\{(V_1, Cov_k(V_1, B_T), B_1), ..., (V_n, Cov_k(V_n, B_T), B_n)\}$ *be a* C*-structure* C *of* B_T *and* σ *be an incision function. The local contraction of* C *by* α*, denoted by* $C \dot{-}_\sigma \alpha$*, is defined as follows:* $C \dot{-}_\sigma \alpha = C \setminus \sigma(B_i \perp \alpha)$*, such that* $V_i \cap \mathcal{V}(\alpha) \neq \emptyset$.

The following theorem characterizes the operation of local contraction with respect to the postulates of kernel contraction defined in [15]:

Theorem 2. *The operation of local contraction* $\dot{-}_\sigma$*, defined above, for a* C*-structure* C *and some incision function satisfies the following postulates:*

- *Success : If* $\alpha \notin Cn(\emptyset)$*, then* $\alpha \notin Cn(C \dot{-}_\sigma \alpha)$.
- *Inclusion :* $C \dot{-}_\sigma \alpha \subseteq C$.

- *Core-retainment : If* $\beta \in (C \setminus C \dot{-}_\sigma \alpha)$*, then there is some* $C' \subseteq C$ *such that* $\alpha \notin Cn(C')$ *and* $\alpha \in Cn(C' \cup \{\beta\})$.
- *Uniformity : If* $\forall C' \subseteq C, \alpha \in Cn(C')$ *iff* $\beta \in Cn(C')$ *then* $C \dot{-}_\sigma \alpha = C \dot{-}_\sigma \beta$.

As a special case, consider a sentence α to contract from C which intersects with several cores of $\mathcal{L}$. How do we choose the sublanguage of C in order to perform the local contraction?

The following proposition shows that, in such cases, we do not need to make a local contraction for all the sublanguages whose core V_i is such as: $V_i \cap \mathcal{V}(\alpha) \neq \emptyset$, but just for one of them.

Proposition 1. *Let* $\{(V_1, Cov_k(V_1, B_T), B_1), ..., (V_n, Cov_k(V_n, B_T), B_n)\}$ *be a* C*-structure* C *of* B_T*, and* $\dot{-}_\sigma$ *be the operator of local contraction determined by an incision function* σ*. Let* α *be a sentence and* $\{V_i, ..., V_j\}$ *be a subset of* $\{V_1, ..., V_n\}$ *such that:* $\forall V_{i'} \in \{V_i, ..., V_j\}$, $V_{i'} \cap \mathcal{V}(\alpha) \neq \emptyset$*. Then:* $C \dot{-}_\sigma \alpha = C \setminus \sigma(B_i \bot \alpha) = ... = C \setminus \sigma(B_j \bot \alpha)$.

In Example 2, if we want to apply local contraction by the sentence $(\neg c \wedge \neg d)$, we notice that $\{c, d\}$ intersects the two cores of the first two structures. In this case, it suffices to proceed only one of the two structures. We take the first structure, for example. We calculate $(\neg c \wedge \neg d)$-kernel set, which is equal to $\{\{\neg c, \neg d\}\}$. By an incision function, we choose the hitting set $\{\neg c\}$ (an arbitrary choice), and we remove it from the first structure, as well as from any structure containing it. Finally, the resulting C-structure is equal to:
$\{(\{a, b, c\}, \{a, b, c, d, e\}, \{a \vee b, b \rightarrow c, \neg d, c \rightarrow (d \vee e)\}), (\{d, e, f\}, \{c, d, e, f, g\}, \{c \rightarrow (d \vee e), \neg d, e \leftrightarrow f, f \rightarrow g\}), (\{g, h, i\}, \{f, g, h, i, j\}, \{f \rightarrow g, \neg g \vee h, i \rightarrow h, i, j \rightarrow i, j\}), (\{j, k, l\}, \{i, j, k, l\}, \{i, j \rightarrow i, j, k \vee l \vee j\})\}$.

The following results show that our local contraction is equivalent to kernel contraction.

Lemma 1. *Let* $\{(V_1, Cov_k(V_1, B_T), B_1), ..., (V_n, Cov_k(V_n, B_T), B_n)\}$ *be a* C*-structure* C *of* B_T*. If* C *is consistent, then:* $B_T \bot \alpha = B_i \bot \alpha$ *such that* $V_i \cap \mathcal{V}(\alpha) \neq \emptyset$.

Corollary 1. *Let* $\{(V_1, Cov_k(V_1, B_T), B_1), ..., (V_n, Cov_k(V_n, B_T), B_n)\}$ *be a* C*-structure* C *of* B_T*,* $-_\sigma$ *be the operator of kernel contraction,* $\dot{-}_\sigma$ *be the operator of local contraction, and let* σ *be an incision function. If* C *is consistent, then:* $C \dot{-}_\sigma \alpha = B_T -_\sigma \alpha$.

This Corollary shows that when contracting a consistent belief base B_T, of a theory T, by a sentence α we can disconsider all sublanguages of a C-structure C of B_T that are not relevant for α[2], since we are sure that the size of the α-kernels of B_T is limited by k. Thus we cannot not see elements of the kernel set of B_T when our research is limited locally (in $\mathcal{L}(Cov_k(V_i, B_T)) \cap B_T$) on one of the sublanguages of C whose core V_i is such as $V_i \cap \mathcal{V}(\alpha) \neq \emptyset$, because the kernel set calculated locally is complete.

[2] A sublanguage $L_i = (V_i, Cov_k(V_i))$ of C is not relevant for a sentence α if $V_i \cap \mathcal{V}(\alpha) = \emptyset$.

Hence, the C-structure model provides us with substantional gains since the size of the set to be updated is always a fraction of the size of the original set and ensures the following result: When we want to contract a piece of information which lies in one of the sublanguage (or straddles only two or three of them) then we can leave most of the C-structure unchanged and contract only one of the affected sublanguages.

5.2 Local External Revision

Finally, our local external revision consists in expanding (locally) the C-structure then the result is contracted by the local contraction operator.

Definition 13. *Let C be a C-structure of a belief base B_T, and $\dot{-}$ be our local contraction operator. Local external revision ($\pm$) consists in extending C by α then to contract the result by $\neg\alpha$: $C \pm \alpha = (C \cup \{\alpha\})\dot{-}\neg\alpha$.*

The following theorem characterizes our local external revision operator with respect to the postulates which should be satisfied by all external revision operators defined in [15]:

Theorem 3. *The operator $\pm$ of local external revision satisfies the following properties for all C-structures C and all sentences α:*

- *If $\neg\alpha \notin Cn(\emptyset)$, then $\neg\alpha \notin Cn(C \pm \alpha)$ (no-contradiction).*
- *$C \pm \alpha \subseteq C \cup \{\alpha\}$ (Inclusion).*
- *If $\beta \in (C \setminus C \pm \alpha)$, then there is some $C' \subseteq C \cup \{\alpha\}$ such that $\neg\alpha \notin Cn(C')$ and $\neg\alpha \in Cn(C' \cup \{\beta\})$ (Core-retainment).*
- *$\alpha \in C \pm \alpha$ (Success).*
- *If α and β are element of C and $\forall C' \subseteq C$, $\neg\alpha \in Cn(C')$ iff $\neg\beta \in Cn(C')$, then $C \cap (C \pm \alpha) = C \cap (C \pm \beta)$ (Weak Uniformity).*
- *$C + \alpha \pm \alpha = C \pm \alpha$ (Pre-expansion).*

Corollary 2. *Let $*$ be an operation of kernel external revision (not-local), let C be a C-structure of B_T, then: $\forall\alpha \in \mathcal{L}, C \pm \alpha = B_T * \alpha$.*

This last Corollary shows that our local revision operator is equivalent to the one defined by Hansson, in [15], which is a *global* revision operator.

6 Complexity Results

Local external revision operator is computationally efficient. Generally we assume that each $\mathcal{L}(Cov_k(V_i, B_T)) \cap B_T$ has relatively small size, say under some fixed p, while the cardinality of B_T (i.e, $|B_T|$) might be quite large. Then given a sentence α with certain number of distinct atoms, local external revision operation by α runs in time which is exponential in p. Thus if p is small compared to $|B_T|$, as is usually the case, the computational cost will be much smaller than that of usual revision operators which are exponential in $|B_T|$.

In [12], the authors were interested in geographical applications revision. They have proposed a revision strategy based on containment property. Indeed, they have opted for a geographical space decomposition into subspaces called blocks, and each block is included in another neighborhood called covering of the block. In this application, the distance for constructing blocks (cores) and coverings of a given geographical space is defined spatially, so it is a real distance. The geographical space is segmented on a set of parcels, and a block is a subset of parcels such that the set of blocks constitutes a partition of the space. Covering of a given block is a subset of parcels surrounding this block for a certain distance which represents the thickness of the covering. To each part, composed of a block and its covering, is attached a spatial knowledge subbase which represents the union of all sentences attached to each parcel belonging to this part.

The theoretical results obtained are encouraging such that the complexity of one of the classical revision operation[3] is proved equals to: $O(|B_T|^3 \times 2^{2\times|B_T|})$. However, that of the strategy based on containment property is equal to: $O(p^3 \times 2^{2\times p})$.

7 Conclusion

There are two principal ways to split up a belief base into several belief subbases. First, it may be done as an addition to the logic by using external information, so that one and the same belief base can be divided into subbases in different ways with keeping classical logic. Secondly, that may be done by deriving the several subbases from the logic, therefore we should define a non-classical inference operation. This case corresponds to what Wassermann and Hansson have done in [19] where they defined an operation of local inference before defining their operators of belief change.

In this paper, we have defined a cutting according to the first approach such as external information was the maximal size of MISs in the real life applications which we have called containment property. Thus we can define an operator of local revision which is at the same time complete and correct. However, if it is rather simple to construct a C-structure corresponding to a given belief base, it is not always possible to make an assumption on the maximal size of MISs. This question is staying open.

The contribution of this paper is twofold. On the one hand, an efficient model has been proposed, allowing one to locate and quantify local inconsistencies (MISs) in sets of propositional sentences (belief bases). We think that such a model should be useful with respect to many domains. For example, this should make the handling of local inconsistencies in the diagnosis domain possible. On the other hand, using the concept of kernel contraction, an operation of local revision has been defined by a new operation of local contraction.

[3] The corresponding revision operator is based on the adaptation of Reiter algorithm. For further detail, either for Reiter algorithm or the corresponding revision operator, see [12].

In future work we intend to carry out a thorough study of this interesting new model for belief representation and belief revision by extending our framework to belief fusion.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. J. Symb. Logic 50, 510–530 (1985)
2. Hansson, S.O.: Reversing the levi identity. Philosophical Logic 22, 637–639 (1992)
3. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates, and counterfactuals, pp. 261–273 (1992)
4. Liberatore, P., Schaerf, M.: The complexity of model checking for belief revision and update. In: AAAI/IAAI, vol. 1, pp. 556–561 (1996)
5. Würbel, E., Papini, O., Jeansoulin, R.: Revision: an application in the framework of gis. In: 7th International Conference on Principles of Knowledge Representation and Reasoning, KR 2000, Breckenridge, Colorado, USA, pp. 505–516 (2000)
6. Parikh, R.: Beliefs, belief revision, and splitting languages. Logic, language and computation 2, 266–278 (1999)
7. Chopra, S., Parikh, R.: An inconsistency tolerant model for belief representation and belief revision. In: IJCAI 1999: Proc. of the 16th International Joint Conference on Artificial Intelligence, pp. 192–199. Morgan Kaufmann Publishers, San Francisco (1999)
8. Chopra, S., Parikh, R.: Relevance sensitive belief structures. Annals of Mathematics and Artificial Intelligence 28, 259–285 (2000)
9. Chopra, S., Georgatos, K., Parikh, R.: Relevance sensitive non-monotonic inference on belief sequences. J. of Applied Non-Classical Logics 11, 131–150 (2001)
10. Kourousias, G., Makinson, D.: Parallel interpolation, splitting, and relevance in belief change. J. Symb. Logic 72, 994–1002 (2007)
11. Peppas, P., Chopra, S., Foo, N.Y.: Distance semantics for relevance-sensitive belief revision. In: 9th International Conference on Principles of Knowledge Representation and Reasoning, KR 2004, Canada, pp. 319–328 (2004)
12. Doukari, O., Jeansoulin, R.: Space-contained conflict revision, for geographic information. In: 10th AGILE International Conference on Geographic Information Science, Aalborg, Danmark (2007)
13. Williams, M.A.: Applications of belief revision. In: ILPS 1997: International Seminar on Logic Databases and the Meaning of Change, Transactions and Change in Logic Databases, London, UK, pp. 287–316. Springer, Heidelberg (1998)
14. Nebel, B.: How hard is it to revise a belief base? In: Dubois, D., Prade, H. (eds.) Handbook of Defeasible Reasoning and Uncertainty Management Systems. Belief Change, vol. 3, pp. 77–145. Kluwer Academic Publishers, Dordrecht (1998)
15. Hansson, S.O.: Kernel contraction. J. Symb. Logic 59, 845–859 (1994)
16. Mazure, B., Saïs, L., Grégoire, E.: Boosting complete techniques thanks to local search methods. Annals of Mathematics and Artificial Intelligence 22, 319–331 (1998)
17. Cook, S.: A short proof of the pigeon hole principle using extended resolution. SIGACT News 8, 28–32 (1976)
18. Chopra, S., Parikh, R., Wassermann, R.: Approximate belief revision. Logic J. IGPL 9, 755–768 (2001)
19. Hansson, S.O., Wassermann, R.: Local change. Studia Logica 70, 49–76 (2002)

Finding Good Starting Points for Solving Nonlinear Constrained Optimization Problems by Parallel Decomposition

Soomin Lee and Benjamin Wah

Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois, Urbana-Champaign
Urbana, IL 61801, USA
{lee203,wah}@illinois.edu

Abstract. In this paper, we develop heuristics for finding good starting points when solving large-scale nonlinear *constrained optimization problems* (COPs) formulated as *nonlinear programming* (NLP) and *mixed-integer NLP* (MINLP). By exploiting the localities of constraints, we first partition each problem by parallel decomposition into subproblems that are related by complicating constraints and complicating variables. We develop heuristics for finding good starting points that are critical for resolving the complicating constraints and variables. In our experimental evaluations of 255 benchmarks, our approach can solve 89.4% of the problems, whereas the best existing solvers can only solve 42.8%.

1 Introduction

The NLP and MINLP problems studied in this paper have nonlinear and non-convex objective and constraint functions that are formulated as follows:

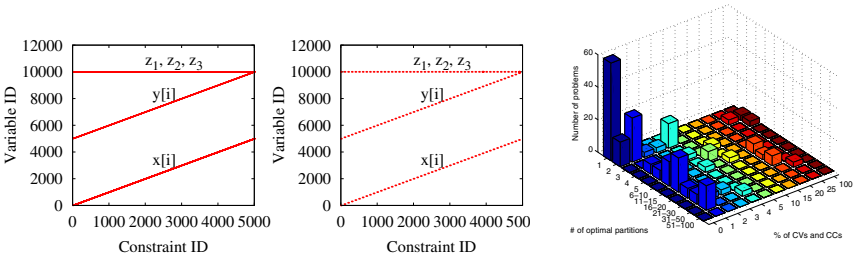
$$(P_m): \qquad \min_{\mathbf{z}} f(\mathbf{z}) \qquad \text{subject to } \mathbf{h}(\mathbf{z}) = \mathbf{0} \text{ and } \mathbf{g}(\mathbf{z}) \leq \mathbf{0}, \tag{1}$$

where $\mathbf{z} = (\mathbf{x}, \mathbf{y})$, and $\mathbf{x} \in \mathbb{R}^v$ and $\mathbf{y} \in \mathbb{D}^v$ are, respectively, vectors of continuous and discrete variables; $f(\mathbf{z})$ is an objective function; $\mathbf{h}(\mathbf{z}) = (h_1(\mathbf{z}), \ldots, h_m(\mathbf{z}))^T$ is a vector of m equality constraint functions; and $\mathbf{g}(\mathbf{z}) = (g_1(\mathbf{z}), \ldots, g_r(\mathbf{z}))^T$ is a vector of r inequality constraint functions. Because no closed form solution to (1) exists, we focus on finding its *constrained local minimum* (CLM) solutions.

Consider ORTHRGDS, an NLP from the CUTE benchmark suite. Its goal is to find values of $x[i]$, $y[i]$, z_1, z_2, and z_3 that minimize $f(\mathbf{z})$ and that satisfy $N = 5000$ non-convex equality constraints $h_i(\mathbf{z}) = 0$ for $i = 1, \ldots, N$:

$$\begin{aligned}
\min_{\mathbf{z}} \ f(\mathbf{z}) &= \sum_{i=1}^{N} \left((x[i] - xd[i])^2 + (y[i] - yd[i])^2 \right) \qquad (2)\\
\text{subject to } h_i(\mathbf{z}) &= 0 \text{ for } 1 \leq i \leq N,\\
\text{where } h_i(\mathbf{z}) &= \left((x[i]-z_1)^2 + (y[i]-z_2)^2 \right)^2 - \left((x[i]-z_1)^2 + (y[i]-z_2)^2 \right)(1+z_3^2)^2,\\
xd[i] &= (C_1 + \cos(2\pi(i-1)/N)) \cos(2\pi(i-1)/N)(1 + C_2 \cos(2\pi C_3(i-1)/N)),\\
yd[i] &= (C_1 + \cos(2\pi(i-1)/N)) \sin(2\pi(i-1)/N)(1 + C_2 \cos(2\pi C_3(i-1)/N)).
\end{aligned}$$

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 65–76, 2008.



a) NLP ORTHRGDS b) 50 sampled constraints c) Localities in 255 benchmarks

Fig. 1. The regular constraint structure of ORTHRGDS and NLP benchmarks studied

Here, $C_1 = 1+1.7^2$, $C_2 = 0.2$, and $C_3 = 237.1531$. The problem cannot be solved by SNOPT [1] and Lancelot, but can be solved by KNITRO and IPOPT [2].

ORTHRGDS belongs a large class of benchmark problems specified by an algebraic modeling language, like AMPL and GAMS, that uses indexes to define groups of variables and constraints with some common properties. Indexing is necessary when specifying large-scale problems because it will be very cumbersome to give a unique name for every variable and constraint. The use of indexed constraints leads to a large number of similar and related constraints.

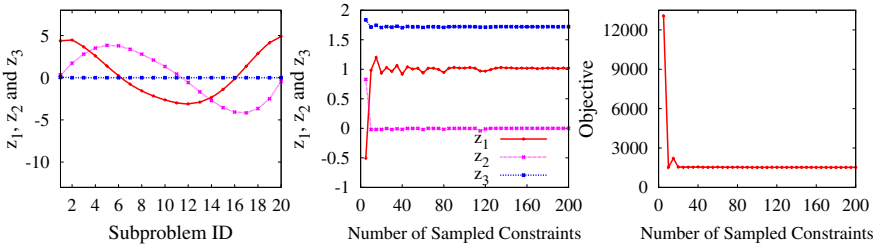
Figure 1a illustrates the regular structure of the constraints in ORTHRGDS. Using the indexed form, it shows a dot where a constraint on the x-axis is related to a variable on the y-axis. In particular, the i^{th} constraint $h_i(\mathbf{z}) = 0$ involves variables $x[i]$ and $y[i]$ and common variables z_1, z_2, and z_3.

Parallel decomposition can be used to exploit the regular constraint structures in large-scale COPs and to decompose them into loosely coupled and much less complex subproblems. The original idea came from solving large-scale linear programming (LP) problems in the 1960s [3]. It partitions the state space of a problem in such a way that the combined state space is the cross product of the subproblem state spaces. The approach leads to the partitioning of variables into possibly overlapping subsets. Those variables that are shared among the subsets are called *complicating variables* (CVs), and those in one subset and not shared are called *local variables* (LVs). As a result of the partitioning of the variables, constraints are also partitioned. Those constraints involving variables in more than one subset are called *complicating constraints* (CCs), whereas those involving local variables in one subset are called *local constraints* (LCs).

A COP has good *localities* when its decomposed subproblems are loosely coupled by a small number of CVs and CCs. For instance, when ORTHRGDS is decomposed by parallel decomposition along the constraint dimension into 20 subproblems, its variables are partitioned into 20 subsets, where the j^{th} subset, $j = 1, \ldots, 20$, has variables $\{x[i], y[i], z_1, z_2, z_3\}$ for $250(j-1)+1 \leq i \leq 250j$. Hence, z_1, z_2, and z_3 are CVs; and all constraints are LCs. If the CVs and CCs can be effectively resolved, then parallel decomposition will be a good method for solving the COP because the complexity of each subproblem is much lower.

Table 1. NLP and MINLP benchmark suites studied in this paper

CUTE	Nonlinear optimization	`http://www.princeton.edu/~rvdb/bench.html`
NLP	Nonlinear programming	`http://plato.asu.edu/ftp/ampl-nlp.html`
NonSys	Nonlinear systems of equations	`http://plato.asu.edu/ftp/ampl_files/nonsys_ampl/`
MacMINLP	AMPL collection of MINLP	`http://www-unix.mcs.anl.gov/~leyffer/macminlp/`
COPS	Large-scale optimization problems	`http://www-unix.mcs.anl.gov/~more/cops/`



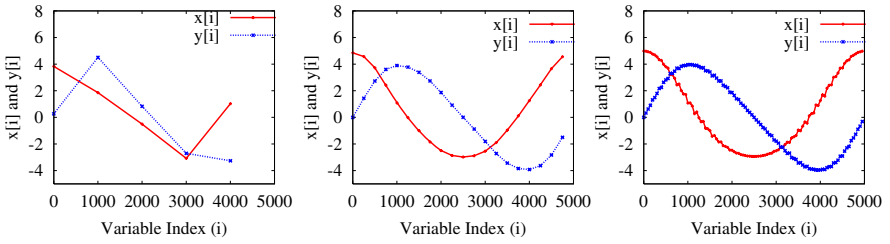
a) Independent evaluations b) Sol. in sampled prob. c) Corresponding Obj. val.

Fig. 2. Convergence behavior when ORTHRGDS is partitioned into 20 subproblems: a) the values of z_1, z_2, and z_3 do not converge when the subproblems are solved iteratively, using the values found in the last problem as the starting point in the next; b) the values of z_1, z_2, z_3 found from the sampled problem; c) the corresponding value of the objective function when fixing z_1, z_2, and z_3 found from the sampled problem

To show that many existing benchmarks have good localities that can be exploited by parallel decomposition, we have exhaustively tested the five benchmark suites in Table 1. After eliminating those unconstrained, linear, or quadratic problems, and small problems with less than 50 variables or constraints, we have 255 benchmarks whose localities are summarized in Figure 1c. The x-axis shows the optimal number of partitions found by the method in Section 2, and the y-axis shows the ratio between the total number of CVs and CCs and the total number of constraints, based on using the optimal number of partitions. Most of the problems have good localities with less than 1% of CVs and CCs. In some problems, the optimal number of partitions is one and no partitioning is needed.

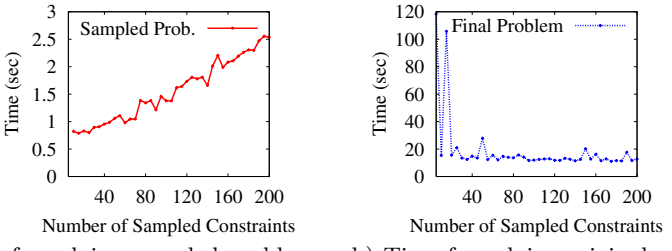
One of the requirements when a COP is partitioned by parallel decomposition is that its CVs and CCs must converge when a solution is found. When the CVs are formulated as CCs, it results in a significant increase in the number of equality constraints that are hard to satisfy. On the other hand, when the subproblems are solved independently and the CVs are propagated from one subproblem to another during the solution process, they will not converge without a proper technique that ensures their consistency across the subproblems.

For example, Figure 2a illustrates that CVs z_i, $i = 1, 2, 3$, in ORTHRGDS will not converge when the z_i found in one subproblem are simply passed to the next as its initial values. The importance of using good starting points is also illustrated in solving EIGENA2 from CUTE. SNOPT can solve EIGENA2 when $q[i, j]$ for $1 \leq i, j \leq 10$, is initialized to an identity matrix (the default), but fails to find a feasible solution when $q[i, j]$ is initialized to a zero matrix.



a) 5 sampled constraints b) 20 sampled constraints c) 100 sampled constraints

Fig. 3. The LVs converge with an increasing number of sampled constraints



a) Time for solving sampled problems b) Time for solving original problem

Fig. 4. Trade-offs between the time for solving a problem with sampled constraints and the time for solving the original problem using the starting point found

The convergence, however, can be significantly improved when the CVs and CCs are initialized properly, and simple techniques are employed to enforce their consistency. In this paper, we show that, for COPs with good structural localities, it is possible to find good starting points by exploiting their structures.

As an illustration, consider a simplified version of ORTHRGDS obtained by uniformly sampling the constraints in Figure 1a. (See Figure 1b when 50 constraints are sampled.) Figure 2b depicts the values of z_1, z_2, and z_3 when using SNOPT to solve ORTHRGDS of different numbers of sampled constraints. Although there are small fluctuations in their values, the CVs converge to some stable values when 20 or more constraints are sampled. This convergence behavior is also extended to LVs. Figure 3 shows the values of LVs $x[i]$ and $y[i]$ for three simplified problems. The graphs show that the values of the variables of the same index are highly related. This *smoothness* property is typical in most of the AMPL benchmarks whose constraints are usually grouped in several forms and the parameters of the constraints in each group are closely related.

Using the starting points found by solving ORTHRGDS with sampled constraints and the other variables interpolated as is shown in Figure 3, we can solve ORTHRGDS by parallel decomposition when we fix the value of its CVs throughout the process. Figure 2c shows that the objective value improves and converges when the number of sampled constraints increases.

In general, our approach of using fixed CVs may not lead to a feasible solution. Section 3 presents a technique that first relaxes the CVs to within some bounds and that gradually tightens the bounds until a feasible solution is found.

There is a trade-off between the time for solving a problem with sampled constraints (which increases as the number of constraints increases−Figure 4a), and the time for solving the original problem using the starting points found (which decreases when the number of sampled constraints increases and the quality of the starting point found improves−Figure 4b). However, the time for solving a sampled problem is usually much smaller. Hence, we can simply choose the number of sampled constraints based on the convergence of the variables found. For example, Figure 2b shows that the solutions of ORTHRGDS converge when 20 or more constraints are sampled.

Our work is based on the theory of extended saddle points and the parallel decomposition algorithm summarized in Section 2 [4]. Our main contribution in this paper is on the development of algorithms for finding good starting points when solving NLPs and MINLPs by parallel decomposition. Without a good starting point, the previous parallel decomposition method gets stuck easily and does not converge.

2 CPOpt for Solving Constraint-Partitioned COPs

We summarize in this section a necessary and sufficient condition on the constrained local minima of P_m [5]. We also describe CPOpt, an implementation for resolving the CCs and CVs when P_m is partitioned by its constraints.

Definition 1. *Let* $a^+ = \max(a, 0)$. *The* penalty function *of* P_m *with penalty vectors* $\boldsymbol{\alpha} \in \mathbb{R}^m$ *and* $\boldsymbol{\beta} \in \mathbb{R}^r$ *is:*

$$L_m(\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{z}) + \boldsymbol{\alpha}^T |h(\mathbf{z})| + \boldsymbol{\beta}^T g(\mathbf{z})^+ = f(\mathbf{z}) + \sum_{i=1}^{m} \alpha_i |h_i(\mathbf{z})| + \sum_{j=1}^{r} \beta_j g_j(\mathbf{z})^+.$$

Without showing the details [5], we have proved that each *constrained local minimum* (CLM) of P_m must satisfy an *extended saddle point condition* (ESPC) and vice versa. Here, an *extended saddle point* (ESP) $(\mathbf{z}^*, \boldsymbol{\alpha}^{**}, \boldsymbol{\beta}^{**})$ is a a local minimum of $L_m(\mathbf{z}, \boldsymbol{\alpha}^{**}, \boldsymbol{\beta}^{**})$ with respect to $\mathbf{z}$ and a local maximum of $L_m(\mathbf{z}^*, \boldsymbol{\alpha}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\alpha}^{**}$ and $\boldsymbol{\beta}^{**}$, when $\boldsymbol{\alpha}^{**}$ and $\boldsymbol{\beta}^{**}$ are non-negative and larger than some thresholds.

ESPC can be applied to solve P_t, a version of P_m whose constraints can be partitioned by parallel decomposition into N subsets:

$$\begin{aligned} (P_t): \quad & \min_{\mathbf{z}(i)} \ f(\mathbf{z}) \\ \text{subject to} \quad & h^{(i)}(\mathbf{z}(i)) = 0, \quad g^{(i)}(\mathbf{z}(i)) \le 0 \quad (\text{LC } i,\ i = 1, \ldots, N) \\ \text{and} \quad & H(\mathbf{z}) = 0, \quad G(\mathbf{z}) \le 0 \quad (\text{CCs}). \end{aligned} \tag{3}$$

The corresponding ESPC can be decomposed into N necessary conditions, one for each subproblem, and another necessary condition on the CCs and CVs. Because finding an ESP of a subproblem is equivalent to solving a MINLP, the ESP search of the i^{th} condition can be formulated as the solution of the following:

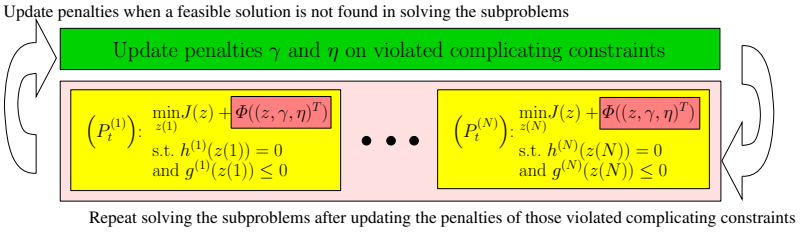


Fig. 5. Partition-and-resolve for finding a constrained local minimum of P_t

$$\left(P_t^{(i)}\right): \min_{\mathbf{z}(i)} f(\mathbf{z}) + \Phi(\mathbf{z}, \boldsymbol{\gamma}, \boldsymbol{\eta}) \text{ s.t. } h^{(i)}(\mathbf{z}(i)) = 0 \text{ and } g^{(i)}(\mathbf{z}(i)) \leq 0, \quad (4)$$

where $\Phi(\mathbf{z}, \boldsymbol{\gamma}, \boldsymbol{\eta}) = \boldsymbol{\gamma}^T |H(\mathbf{z})| + \boldsymbol{\eta}^T G(\mathbf{z})^+$ is the weighted sum of the CC functions. For simplicity, the CVs in (3) have been formulated as CCs in (4).

Figure 5 illustrates the concept. After decomposing P_t into N subproblems, the process iterates between solving the subproblems and re-weighting those violated CCs until a feasible solution to P_t is found.

Next, we summarize the design of CPOpt [4], which implements the approach in Figure 5. CPOpt has the following steps.

a) It first analyzes a COP represented in AMPL and decomposes its constraints by their localities in order to minimize the number of CCs across the subproblems. An *index vector* in an AMPL model is a finite ordered array of discrete elements that are used to index variables and constraints. This convenient representation allows a suitable partitioning of the constraints to be found by enumerating the different combinations of index vectors. Another benefit is that the overhead of finding the partitions is much smaller than that of clustering the constraints because the number of index vectors is very small (usually ≤ 5).

b) CPOpt find the proper granularity of partitioning by an iterative procedure. A *partitioning index vector* (PIV) of an AMPL model is defined to be an index vector used for partitioning the constraints. Using a PIV, an N partition is a collection of the disjoint subsets of the PIV, $S_1, \cdots, S_N$, where PIV $= S_1 \cup \cdots \cup S_N$. In general, the Cartesian products of the PIVs can be partitioned along one or more vectors into subsets. An interesting property observed from the benchmarks tested is that the best combination of the PIVs for a problem instance that leads to the minimum number of CCs is independent of the number of partitions N [4]. Hence, to determine the best combination, all possible combinations of PIVs are enumerated by fixing an arbitrary value of N.

Based on the PIVs selected and a convex relationship between the number of partitions and the total solution time, CPOpt finds the optimal number of partitions by a binary search. Starting from a large number of partitions, one subproblem in the set is solved to estimate the overhead of solving all the subproblems in the set, while ignoring the overhead of resolving the CCs (a lower-bound estimate). Then, the number of partitions is reduced by half and the procedure repeated until the estimated overhead starts to increase. The number of partitions found at that point is very close to the optimal number [4].

c) CPOpt then calls an existing solver to solve each partitioned subproblem and resolves those violated CCs iteratively by updating their penalties. After solving each subproblem, γ and η on the violated CCs are increased by some constant rates. To avoid getting stuck in infeasible stationary points of the penalty function, periodic decreases of γ and η are allowed, which "lower" the barrier in the penalty function and allow local descents with respect to z to escape from an infeasible region. The process is repeated until all the constraints are satisfied or the maximum number of iterations is exceeded.

One of the difficulties encountered in CPOpt is that the PIVs alone may not fully exploit the constraint locality of a problem. For example, the benchmark WM_CFy from the NLP suite has 8,709 variables and 11,338 constraints and cannot be solved by SNOPT in over 3,600 sec. The best PIV leads to 2,646 CCs, which are too many to be handled. A close observation of its CCs leads to some common expressions and operators that can be further partitioned into multiple LCs with a small number of CVs. For example, when the constraints are partitioned by PIV $i = \{1, \ldots, 2N\}$ into two disjoint subsets $S_1 = \{1, \ldots, N\}$ and $S_2 = \{N+1, \ldots, 2N\}$, the CC $\left(\max_{i=1,\ldots,2N} x_i\right)/\sqrt{\sum_{i=1}^{2N} y_i^2} \leq 0$ can be rewritten into $\max\{\left(\max_{i=1,\ldots,N} x_i\right), t_{2,1}\}/\sqrt{\sum_{i=1}^{N} y_i^2 + t_{2,2}} \leq 0$. This is an LC in the first subproblem with two CVs $t_{2,1} = \max_{i=N+1,\ldots,2N} x_i$ and $t_{2,2} = \sum_{i=N+1}^{2N} y_i^2$ computed in the second subproblem. A similar LC can be constructed in the second subproblem using two CVs in the first subproblem. When WM_CFy is partitioned into six subproblems in this way, each can be solved in 5 sec. by SNOPT.

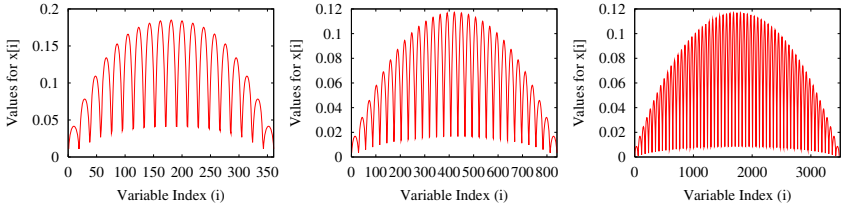
3 Generation of Good Initial Starting Points

Our goal is to derive a simplified version of the original COP whose solution can be extended to provide a good starting point. This is done by solving one or more simplified versions of the COP and by generalizing the solutions found. In this section, we present three methods based on the problem characteristics.

a) Constraint sampling and interpolations. For NLPs whose constraints are independent except for a few global shared variables, the constraints can be uniformly sampled in order to generate one or more simpler versions of the original NLP that are coupled through the shared variables. Using the approach described in Section 1 for solving ORTHRGDS, the NLPs with sampled constraints are solved until the shared variables converge.

To avoid enumerating all possible combinations of sampled constraints, we determine the number of sampled constraints as follows. As is depicted in Figure 2b, the CVs converge to some stable values as more constraints are sampled. Therefore, we start solving a simplified problem using a very small number (N_1) of sampled constraints. We then gradually increase the number of sampled constraints (N_i) until some convergence is observed. Let $\mathbf{v}_i$ be the vector containing the values of the CVs after solving a simplified problem with N_i sampled constraints. Convergence is then determined as follows:

$$(1-a)|\mathbf{v}_{i-1}| - K \quad \leq \quad |\mathbf{v}_i| \quad \leq \quad (1+a)|\mathbf{v}_{i-1}| + K, \quad 0 < a < 1, \qquad (5)$$



a) EX1 ($n = 19$) b) EX1 ($n = 29$) c) Generalized EX1 ($n = 59$)

Fig. 6. The common features of the solutions of three simplified versions of EX1

where K is a constant in case $|\mathbf{v}_i|$ is very close to zero. If $|\mathbf{v}_i|$ is within the bound $(1 \pm a)|\mathbf{v}_{i-1}|$ or if N_i exceeds 10% of the total constraints, then we stop and use $\mathbf{v}_i$ as the starting point. This approach is intuitively sound, as we expect $\mathbf{v}_i$ to converge to some stable form as more constraints are sampled.

Using the values of variables solved in the simplified NLP, we interpolate the values of variables in the original NLP that are not solved in the simplified version into a starting point. While fixing the CVs, we then solve the original NLP by parallel decomposition. Since no feasible solution may be found when the CVs are fixed, we relax the CVs so that they can be solved within some bounds. Once a feasible solution is found, we gradually tighten the bounds until the CVs are consistent across the subproblems. Overall, the approach is used in solving 34 (or 13.3%) of the 255 benchmarks tested (indicated by A in Table 2).

b) Constraint sampling and extrapolation. For an NLP whose constraints are related to their neighboring constraints by some common variables, uniformly sampling the constraints may result in a new problem with independent constraints that do not have common variables. For example, the problem DTOC6 from CUTE has constraints of the form $y[i] - y[i+1] + \exp(x[i]) = 0$ for $1 \leq i \leq 5000$, and the i^{th} constraint is related to the $i+1^{\text{st}}$ constraint by $y[i+1]$. This means that if every other constraint is sampled, then the resulting problem has independent constraints. In this case, solving the simplified problem will not lead to any insights on the starting point of the original problem. For these NLPs, instead of uniformly sampling the constraints, we generate a version using a subset of their contiguous constraints and eliminate those variables that are not in the sampled constraints. For instance, in DTOC6, we generate a problem with 500 constraints that are related to variables $y[1], \ldots, y[500]$.

After solving the simplified problem, we extrapolate the values of the variables found to those in the original problem. In the simplest case, the extrapolations can be done by assigning the value found for a variable in the simplified problem to the corresponding variable in the original problem and by interpolating the values of other variables. This works in DTOC6 when we assign $y[j]$ in the sampled problem to $y[j']$, $j' = 10(j-1)+1$, in the original problem. This is used in solving 128 (or 50.2%) of the benchmarks (indicated by B1 in Table 2).

In more general cases, the solution of the simplified problem may be in a complex form, although the solution can be generalized using some features to be identified. For example, EX1 of the NLP suite has variables $x[i]$, $1 \leq i \leq n^2$ and $n = 59$. When we solved three simplified problems with $n = 19,\ 29,\ 39$, the solutions have, respectively, 19, 29, and 39 peaks with a similar envelope (Figures 6a and 6b). Hence, we generalize the initial solution of EX1 to have 59 peaks and the same envelope as that of $n = 39$ (Figure 6c). This technique is used in solving 36 (or 14.1%) of the benchmarks (indicated by B2 in Table 2).

c) Constraint relaxation for MINLPs. For MINLPs, we generate a good starting point by solving it as an NLP without the integrality requirement. We further apply the two approaches above in case that the relaxed MINLP is too large to be handled by an existing NLP solver. This technique is used in solving 30 (or 11.8%) of the benchmarks (indicated by C in Table 2).

To effectively analyze the constraint structure of a COP written in AMPL, we have developed a recursive descent parser (using the Perl module *ParseRecDescent* from `www.cpan.org`) that can automatically parse its representation and identify the PIVs and the variables in the constraints. The automated CPOpt rewrites the original AMPL program into subproblems in AMPL, generates one or more AMPL programs with sampled constraints, solves the sampled problems to generate good starting points, solves some subproblems in order to determine the optimal granularity, and implements the approach in Section 2 to find CLMs.

4 Analysis of the Starting Points Found

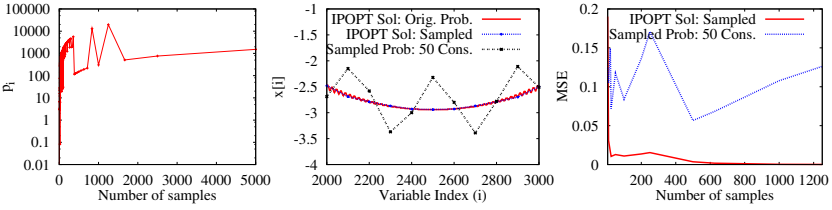
Due to space limitation, we informally analyze the quality of the starting points.

a) Convergence of the objective function. Let p_i^* be the global minimum of P_i, a problem with i sampled constraints, and p_N^* (or p^*) be the global minimum of P_N, the original problem with N constraints. Obviously, $p_i^* \leq p_j^*$ for $i \leq j$, where the constraints of P_i is a subset of those of P_j. Therefore, the optimal value of each simplified problem is a lower bound for p_N^*. That is,

$$p_0^* \leq p_1^* \leq \ldots \leq p_i^* \leq \ldots \leq p_j^* \leq \ldots \leq p_N^*. \tag{6}$$

Let $p_i \geq p_i^*$ be a local minimum solution of P_i found by some solver. Unlike (6), there is no similar relation between p_i and p_j when the constraints of P_i is a subset of those of P_j because p_i may be greater than or smaller than p_j. Figure 7a illustrates this fact that the solutions of the sampled problems found by IPOPT [2] do not increase monotonically as the number of constraints grows.

b) Smoothness and convergence of the starting points. In generating starting points for regularly indexed problems, we assume that the values of variables of the same index are highly related. Those variables have a *smoothness* property when the mean squared error (MSE) between their optimal and the linearly interpolated values decreases monotonically as the number of sampled variables increases beyond some threshold. As is discussed in Section 1, this smoothness property has been found in many AMPL benchmarks, where each problem has constraints that are grouped and the parameters in each group are closely related.



a) Objective value p_i b) Variable $x[i]$ c) MSE of starting points

Fig. 7. The convergence of the objective values and the MSE of sampled solutions

With the smoothness property, our proposed techniques in Section 3 generally work well because they are based on linear interpolations.

Figure 7b illustrates a) the values of $x[i]$, $2001 \leq i \leq 3000$, found by IPOPT (IPOPT Sol: Orig. Prob.), b) the linearly interpolated values of 50 sampled final solutions (IPOPT Sol: Sampled), and c) the variable values found by solving a problem with 50 sampled constraints (Sampled Prob: 50 Cons). Figure 7c shows that the MSE between (a) and (b) for $1 \leq i \leq 5000$ decreases monotonically when the number of samples exceeds 250 and converges to nearly zero when the number of samples is 500. In contrast, the MSE between (a) and (c) oscillates around 0.1, and the MSE between (a) and the default starting point is 8.0664.

5 Experimental Results

We compare the performance of CPOpt to that of other leading solvers, using the 255 NLP and MINLP benchmarks described in Section 1.

Table 2 summarizes those problems that can be solved by CPOpt but cannot be solved by SNOPT (an NLP solver using SQP) [1] and FilMINT (a MINLP solver using branch-and-cut and filterSQP). These are generally large problems, and CPOpt can solve each in under 5 minutes (33 minutes for space-960)

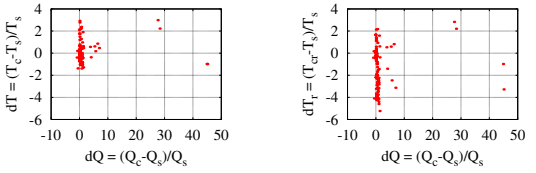
However, there are 27 (or 10.6%) benchmarks that cannot be solved by CPOpt. In particular, LUKVLE2 from the NLP suite has a poor locality. The best PIV leads to $10(N-1)$ CVs when partitioned into N subproblems, which are too many to be handled efficiently. All the other problems (LUKVLI2, TOP1-15x05, TOP1-30x10, TOP1-60x20, CATMIX400) not solvable by CPOpt have similar characteristics. There are also 21 problems that cannot be solved by SNOPT even for significantly simplified versions.[1]

Figure 8 compares the normalized time and quality of the 109 problems that can be solved by CPOpt as well as SNOPT or FilMINT. These problems are smaller problems that require very small amount of execution times. In general,

[1] Problems not solvable by SNOPT, even for simplified versions, include EX6, CONT6, CONT7, CONT8, CONT5_1, CONT5_2_1, CONT5_2_2, CONT5_2_3, CONT5_2_4, CONT5_400, TWOD, POROUS1, POROUS1_L, SEMICON1, SEMICON1_L, SEMICON2, LUKVLE4, LUKVLE6, LUKVLI6, TETRA4, TETRA5.

Table 2. Result on solving NLP and MINLP benchmark suites in Table 1. The experiments were performed on an AMD 2-GHz computer with RedHat Linux 3.4.6 and 2 GB memory. For each instance, $n_c(*)$, n_v, and n_p are, respectively, the number of (*nonlinear) constraints, variables, and partitions; and n_s is the number of sampled constraints (for technique A), or the number of variables in the simplified problem (for B), or nil (for C where MINLP is relaxed to NLP). Tech. refers to the method in Section 3 for generating starting points; and Iter. is the number of iterations for resolving the violated CCs.

Benchmark	$n_c(*)$	n_v	n_p	n_s	Tech.	Iter.	Time	Sol	Benchmark	$n_c(*)$	n_v	n_p	n_s	Tech.	Iter.	Time	Sol
CUTE (Not Solvable by SNOPT in 3,600 sec.)							CPOpt		CUTE (Not Solvable by SNOPT in 3,600 sec.)							CPOpt	
dtoc1l	9990(0)	14985	10	1495	B1	1	22.50	155.69	dtoc2	3996(3996)	5994	10	598	B1	1	7.65	0.51
dtoc4	9997(4997)	14996	10	749	B1	1	18.27	10.74	dtoc6	5000(5000)	10000	10	1001	B1	1	40.75	6.76e5
hager2	5000(0)	10000	20	201	B1	1	10.17	0.00	hager4	5000(0)	10000	20	1001	B1	1	11.66	3.46
orthrdm2	2000(2000)	4003	10	10	A	1	10.13	155.96	orthregd	5000(5000)	10003	25	50	A	1	23.36	1524.21
orthrgdm	5000(5000)	10003	20	50	A	1	20.47	1513.99	orthrgds	5000(5000)	10003	20	20	A	1	18.58	1545.35
catenary	166(166)	496	15	54	B1	1	4.95	-5.45e5	dtoc5	4999(4998)	9998	20	999	B1	1	14.54	1.50
hadamard	256(128)	65	1	16	B1	1	5.98	1.01	kissing	903(903)	127	1	36	B1	1	65.55	1.20
orthregc	5000(5000)	10005	20	50	A	1	19.72	190.79									
NLP (Not Solvable by SNOPT in 3,600 sec.)							CPOpt		NLP (Not Solvable by SNOPT in 3,600 sec.)							CPOpt	
cont5	3717(236)	3953	1	517	B1	1	13.95	0.55	ex4	3717(3481)	7198	1	802	B2	1	8.45	0.08
lukvle1	49998(49998)	50000	10	500	B2	1	22.31	6.23	lukvle3	2(2)	50002	1	502	B2	1	3.14	65.12
lukvle5	49996(49996)	50002	100	502	B2	1	203.67	0.37	lukvle7	4(4)	50002	100	502	B2	1	351.98	-6.61e4
lukvle8	49998(49998)	50000	2	1000	B2	1	6.41	4.13e6	lukvle9	6(6)	50000	100	50	B2	1	366.62	4994.66
lukvle10	49998(49998)	50000	2	500	B2	1	6.35	1.77e4	lukvle11	33330(33330)	49997	2	497	B2	1	4.83	0
lukvle12	37497(37497)	49997	50	997	B2	1	123.42	7.72e4	lukvle13	33330(33330)	49997	40	497	B2	1	210.26	4.02e5
lukvle14	33330(33330)	49997	40	497	B2	1	225.75	3.80e5	lukvle15	37497(37497)	49997	2	997	B2	1	4.61	0
lukvle16	37497(37497)	49997	2	997	B2	1	4.39	0	lukvle17	37497(37497)	49997	10	997	B2	1	22.71	7.14e4
lukvle18	37497(37497)	49997	5	997	B2	1	12.96	6.00e4	lukvli1	49998(49998)	50000	2	50	B2	1	6.30	0
lukvli3	2(2)	50000	1	500	B2	1	3.06	11.58	lukvli4	49998(49998)	50000	100	500	B2	1	369.26	2.01e4
lukvli5	49996(49996)	50000	2	500	B2	1	10.28	0.37	lukvli7	4(4)	50000	100	500	B2	1	365.34	-1.86e4
lukvli8	49998(49998)	50000	2	500	B2	1	8.75	4.13e-6	lukvli9	6(6)	50000	100	50	B2	1	353.80	4994.67
lukvli10	49998(49998)	50000	2	500	B2	1	8.69	1.77e4	lukvli11	33330(33330)	49997	2	497	B2	1	6.48	0
lukvli12	37497(37497)	49997	10	997	B2	1	19.64	0	lukvli13	33330(33330)	49997	3	497	B2	1	6.98	1.08e-8
lukvli14	33330(33330)	49997	40	497	B2	1	264.02	7.10e4	lukvli15	37497(37497)	49997	2	997	B2	1	4.62	0
lukvli16	37497(37497)	49997	2	997	B2	1	4.60	0	lukvli17	37497(37497)	49997	100	997	B2	1	255.70	1.92e-5
lukvli18	37497(37497)	49997	2	997	B2	1	4.49	0	WM_CFy	11338(2268)	8709	6	1184	B1	1	6.25	1.25
Nonsys (Not Solvable by SNOPT in 3,600 sec.)							CPOpt		Nonsys (Not Solvable by SNOPT in 3,600 sec.)							CPOpt	
bratu2d	61504(61504)	62500	1	10000	B1	1	26.02	0	bratu2d_l	248004(248004)	250000	1	10000	B1	1	23.53	0
bratu2dt	61504(61504)	62500	1	10000	B1	1	26.27	0	bratu2dt_l	248004(248004)	250000	1	10000	B1	1	26.27	0
bratu3d	27000(27000)	32768	1	512	B1	1	2.79	0	cbratu2d	123008(123008)	62500	1	10000	B1	1	59.69	0
cbratu2d_l	316808(316808)	160000	1	10000	B1	1	55.38	0	cbratu3d	11664(11664)	16000	1	125	B1	1	1.99	0
chemrctb	50000(49998)	50000	1	500	B1	1	2.53	0	porous2	61504(61504)	62500	1	10000	B1	1	29.29	0
porous2_l	248004(248004)	250000	1	10000	B1	1	29.34	0	semicon1	50000(50000)	50002	2	500	B1	1	30.32	0
MacMINLP (Not Solvable by FilMINT in 3,600 sec.)							CPOpt		MacMINLP (Not Solvable by FilMINT in 3,600 sec.)							CPOpt	
c-reload-q-104	3338(3130)	12906	13		C	42	360.60	-1.24	c-reload-q-49	1430(1332)	3292	7		C	10	68.43	-1.14
space-960-ir	3617(960)	2657	30		C	21	170.08	7.70e6	space-960	8417(960)	15137	30		C	20	2100.08	7.84e6
stockcycle	97(0)	480	6		C	2	5.90	1.20e5	trimlon6	72(12)	168	6		C	5	18.09	15.6
trimlon12	72(12)	168	12		C	47	347.27	98.7	trimloss5	90(5)	161	5		C	6	69.88	10.7
trimloss6	154(7)	345	6		C	8	72.94	22.15	trimloss7	154(7)	345	7		C	4	57.30	26.80
trimloss12	372(12)	800	12		C	52	348.98	150.85									



a) Total time vs quality b) Resolution time only vs quality

Fig. 8. Normalized time and quality of the 109 problems that can be solved by CPOpt as well as SNOPT or FilMINT

due to the additional overheads for partitioning, finding good starting points, and resolving violated CCs, CPOpt is slower in solving these problems (Figure 8a). However, if only the times for resolution are considered, then CPOpt is much faster in solving most of them (Figure 8b). Note that the normalized quality (dQ) of CPOpt for four problems is much worse when the solutions found by SNOPT or FilMINT (Q_s) are very close to zero.

References

1. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM review 47(1), 99–131 (2005)
2. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Programming 106(1), 25–57 (2006)
3. Dantzig, G.B., Wolfe, P.: Decomposition principle for linear programming. Operations Research 8, 101–111 (1960)
4. Wah, B.W., Chen, Y.X.: Solving large-scale nonlinear programming problems by constraint partitioning. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709, pp. 697–711. Springer, Heidelberg (2005)
5. Wah, B., Chen, Y.X.: Constraint partitioning in penalty formulations for solving temporal planning problems. Artificial Intelligence 170(3), 187–231 (2006)

An Early Warning System for the Prediction of Criminal Careers

Tim K. Cocx, Walter A. Kosters, and Jeroen F.J. Laros

Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
tcocx@liacs.nl

Abstract. Dismantling networks of career criminals is one of the focus points of modern police forces. A key factor within this area of law enforcement is the accumulation of delinquents at the bottom of the criminal hierarchy. A deployed early warning system could benefit the cause by supplying an automated alarm after every apprehension, sounding when this perpetrator is likely to become a career criminal. Such a system can easily be built upon existing, strategic, analysis already performed at headquarters. We propose a tool that superimposes a 2-dimensional extrapolation on a static clustering, that describes the movement in time of an offender through the criminal spectrum. Using this extrapolation, possible future attributes are calculated and the criminal is classified accordingly. If the predicted class falls within the danger category, the system notifies police officials. We outline the implementation of such a tool and highlight test results on the Dutch National Criminal Record Database. Certain problematic situations, like time constraints, privacy concerns and reliability issues, are also discussed.

1 Introduction

One of the common factors in today's formation of policy is that of prediction. Usually, the ability to successfully estimate future behavior of individuals, has a fundamental role in the development of any corporate (communication) strategy. Prediction of later revenues might safeguard investments in the present. Large corporations invest heavily in this kind of activity to help focus attention on possible events, risks and business opportunities. Such work brings together all available past and current data, as a basis on which to develop reasonable expectations about the future. The increased availability of useful data has led to the development of more automated processes that enable the acquirement of such prognoses by constructing knowledge from different digital sources.

Current and future behavior are regularly described by employing a number of different algorithmic computer constructs. This branch of data mining, known as *predictive modeling*, provides predictions of future events and may be transparent and readable in for example *rule based systems* and opaque in others such as *neural networks*. Usually, *detection theory* [1] serves as a foundation whereupon these models are selected. They often employ a set of classifiers to determine the probability of a certain item belonging to a dataset, like, for example, the

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 77–89, 2008.

probability of a certain email belonging to the subset "spam". These methods are well suited to the task of predicting certain unknown attributes of an individual by analyzing the available attributes, for example, estimating the groceries an individual will buy by analyzing demographic data. It might, however, also be of interest to predict shopping behavior based upon past buying behavior alone, thus predicting the future development of a certain sequence of recent events. Examples of this are the prediction of animal behavior when their habitats undergo severe changes, in accordance with already realized changed behavioral patterns, or the prediction of criminal careers based upon earlier felonies.

This paper discusses a new tool that attempts to predict the continuation of individual criminal careers: the criminal activities that a single individual exhibits throughout his or her life. The national police annually extracts information from digital narrative reports stored throughout the individual departments and compiles this data into a large and reasonably clean database that contains all criminal records from the last decades. Analysis of this database leads to a clustering of careers, that yields important information for the formation of strategic policies [8]. Furthermore, the clustering can serve as a basis on which to track the movement in time of a certain perpetrator. A plotted line though the first few years of such a career could potentially be extended and a future class could be assigned to such an individual. Integration of this toolset into police software enables the automatic prediction of a criminal career each time a new entry for an offender is submitted. If the calculated career then falls within a preconfigured set of danger categories, an early warning will be sent to police officers in charge of for example organized crime, and action can be taken accordingly. In this paper, we discuss the challenges in career prediction and the specific problems that arise with the implementation of software that transfers higher levels of knowledge discovery to prediction of individual cases.

2 Background

The number of data mining projects in the law enforcement area is slowly increasing. Both inside and outside of the academic world large scale projects are underway. In this section we discuss related work and provide an overview of our approach.

2.1 Related Work

One of the larger academic projects, known as COPLINK, is a police-university collaboration in Arizona, where work has been done in the exploitation of data mining for cooperation purposes [4], the field of entity extraction from narrative reports [5], and social network analysis [16,6]. The FLINTS project and FinCEN [12] aim at revealing links between crimes and criminals and to reveal money laundering networks by comparing financial transactions. Also, Oatly et al. [14] linked burglary cases in the OVER project. Clustering techniques are widely used in the law enforcement arena as well, like for example by Adderly and Musgrove

[2], who applied clustering techniques and Self Organizing Maps to model the behavior of sex-offenders, and by Cocx et al. [7,8] who made an attempt at clustering criminal investigations to reveal what offenses were committed by the same group of criminals, revealed links between crimes and demographic data and the above mentioned automated analysis of criminal careers.

2.2 Criminal Careers

Criminal careers have always been modeled through the observation of specific groups of criminals. A more individually oriented approach was suggested by Blumstein et al. [3]: little definitive knowledge had been developed that could be applied to prevent crime or to develop efficient policies for reacting to crime until the development of the criminal career paradigm. A criminal career is the characterization of a longitudinal sequence of crimes committed by an individual offender. Participation in criminal activities is obviously restricted to a subset of the population, but by focusing on the subset of citizens who do become offenders, they looked at the *frequency*, *seriousness* and *duration* of their careers. Blumstein et al. [3] also look at one specific type of career, that of career criminals; offenders who make crime their life-time profession. A distinction can be made between minor and heavy career criminals, respectively those who are guilty of petty crimes, like theft or fencing, most often the result of addiction disorders, and those involved in more severe felonies, often through membership in a crime syndicate.

The dismantlement of crime syndicates ranks seventh in the current list of top priorities of the FBI [11]. As the growth of crime syndicates starts at the bottom layer of the criminal hierarchy, which is most often shaded to law enforcement agencies, a tool that makes educated guesses about people who are most likely to enter such organizations can be a valuable asset.

Analysis. The analysis of criminal careers is usually done through the extraction and aggregation of knowledge from a criminal record database, commonly containing an unexpectedly large portion of the population (6% in the Netherlands). It can best be accomplished by adapting a *multiset* approach. A multiset is a collection where each element can occur more than once. The set of all distinct elements in that multiset is called its *underlying set*. Figure 1 describes the relation between the two and shows how we employ it to represent a criminal's activities in a single year.

The multiset representation offers advantages, most notably the availability of standard approaches to compare multisets and calculate distances between them. Kosters and Laros [13] devised a distance function for multisets that generalizes well-known distance measures and allows for the incorporation of weights for an element, e.g., element x counts twice as much as element y. This metric contains a customizable function f that can be adapted to fit specific knowledge domains. A tailored f was developed that calculates the distance between two crime-multisets.

Instead of a strict number-wise comparison between years (comparing the first year of criminal a with the first year of criminal b, the second year of a with the

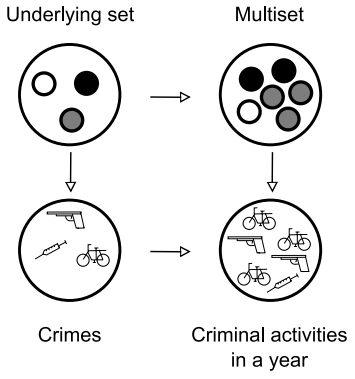


Fig. 1. A multiset representation of a criminal profile in a single year

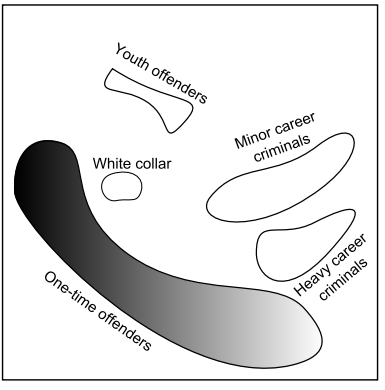


Fig. 2. Impression of a clustering of criminal careers

second year of b, etc.) a novel *alignment* of the mentioned multisets was proposed. This method strives for an optimal automated matching of years, using the distance measure described above, dealing penalties for every mutation needed, which enables a police analyst to better cope with situations like captivity, forced inactivity or unpenalized behavior. Figure 3 shows the intuition behind such a method.

Based upon the distance between two aligned criminal careers, a distance matrix was constructed, containing the distances between all couples of criminal careers. This matrix was then visualized into a 2-dimensional clustering using some kind of *Multi-Dimensional Scaling* [10]. An impression of the results can be seen in Figure 2.

The ultimate goal, predicting if certain offenders are likely to become (heavy) career criminals, was not realized or elaborated upon. Further investigation of this possibility led to the need of a good 2-dimensional visual *extrapolation* system.

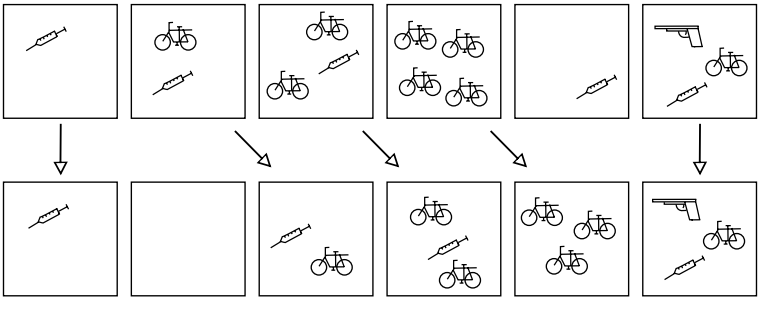


Fig. 3. Two criminal careers who's similarity is revealed by alignment

2.3 2-Dimensional Extrapolation

A new way of predicting the "movement in time" of items through predefined classes by analyzing their changing placement within a static, preconstructed 2-dimensional clustering of other individuals was discussed in [9]. It employs the visualization realized in previous steps within item analysis, rather than performing complex calculations on each attribute of each item. For this purpose a range of well-known mathematical extrapolation methods was adopted that were adapted to fit the need for 2-dimensional extrapolation.

As a first step in this paradigm, the individual sequence to be extrapolated is selected and the sequence up to that moment is calculated for each time unit. Then, the distance between all time units and all other sequences already in the clustering is calculated. Each time-unit is then clustered in the original clustering, leaving all the existing elements in their original location. Note that this requires an iterative clustering method, like for example a randomized push and pull algorithm, rather than a Multi-Dimensional Scaling technique that calculates the entire visualization in a single pass. The resulting coordinates for each time unit can now be utilized by an extrapolation scheme.

The visual extrapolation paradigm has two distinguished advantages. On one hand, the results can immediately be visualized to the end-user, also enabling the user to derive how the results were reached in the first place, on the other hand, the computational complexity is very low, requiring only a few distances to be calculated and only a few elements to be plotted within an existing clustering. Note that the calculated x and y coordinates have no actual meaning; they serve only to give an idea where the item under consideration will be displayed relative to existing elements (initially also positioned on an arbitrary location).

The approach offers several different extrapolation schemes that are suitable for usage within a plane: second or third degree polynomial extrapolation, an x,y system with second or third degree polynomial extrapolation or spline extrapolation with straight line or polynomial continuation. Depending on the task domain the one with the best results should be selected. More information on visual extrapolation and the different extrapolation schemes can be found in [9].

3 Approach

The incorporation of a prediction tool into regular, police software comes with some serious time constraints; the early warning system should obviously not interfere with regular daily operations. Hence, the computational complexity of a prognosis tool should be minimal. Standard, mathematical, extrapolation methods that, for example, extrapolate every attribute separately, have difficulties complying with this demand. Next to that fact, a series of 0's will always be extrapolated to 0 by standard approaches. Some crimes, however, tend to have the property that they are far more likely to be committed after a few years of criminal activity, effectively creating a series of 0's that needs to be extrapolated by a number other than 0. This effectively renders standard extrapolation methods useless. Using the clustering as a depiction of domain knowledge, this

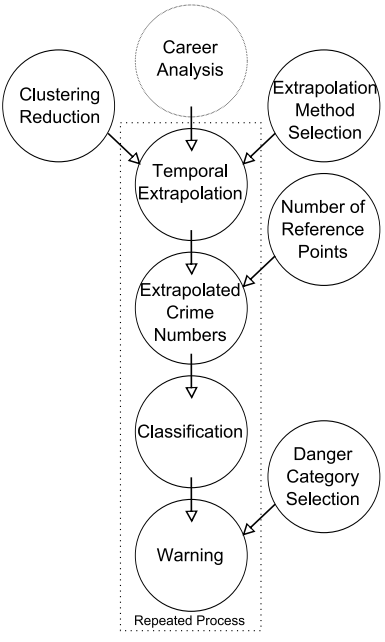


Fig. 4. Approach for criminal career prediction

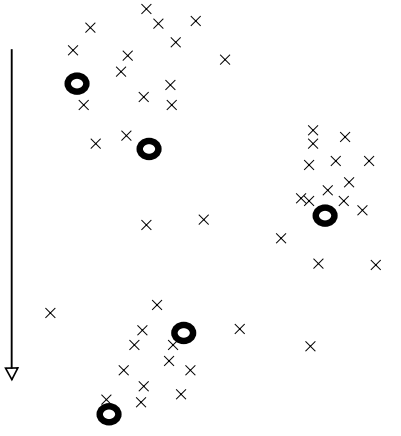


Fig. 5. Example for clustering reduction; only the circles are kept

problem can be dealt with effectively. We therefore resort to a more knowledge discovery oriented approach, specifically the 2-dimensional extrapolation of clustering results mentioned above. According to [9] the power of a clustering visualization resulting from career comparison can easily be used to reach accurate results without an abundance of computations. Using the temporal extrapolation method, only the *coordinates* of the careers in the clustering are used for the largest part of the algorithm. In Figure 4 we show the steps we take to come from the mentioned clustering to a decision on issuing a warning. Each step is described below.

Within this figure, only the boxed steps are taken every time the method is used to determine the development of a single career. The other steps are taken beforehand (the clustering) or are predetermined by research (selection of extrapolation method, clustering reduction and reference points) or by domain experts (selection of danger categories).

3.1 Clustering Reduction and Extrapolation Selection

Although the speed of the temporal extrapolation scheme is very high, the accuracy of the results can vary with the selection of a specific extrapolation method. It is obviously important to determine the optimal option for extrapolation through field testing, which is explored in the experiments in Section 4.

Given the size of a typical criminal record database, ranging in the millions, a significant gain in speed could be realized by reducing the amount of offenders within the clustering. Naturally, care must be taken to realize this decrease without sacrificing the descriptiveness of the clustering itself, for example because certain clusters might lose enough "members" to cause a substantial reduction in "attraction" to newly entered individuals. Within our approach, the reduction is realized using a top down approach (as seen in Figure 5), that deletes items from a y-coordinate sorted list, keeping only every tenth individual.

This will reduce the amount of individuals with a factor 10, retaining a higher similarity with the original clustering than what would be the case if just the first tenth of the original database was used. The rationale behind this is that using this method, the same amount of individuals will be removed from every "height" in the image, preserving the shape of the image as much as possible. A strong argument against simple database removal is the fact that the database could be sorted in many ways, both implicitly and explicitly, without the user's knowledge. Therefore, removal of specific parts of the database could have unintended effects on the outcome, creating a "polluted" clustering. If necessary this process can be repeated to create even smaller clustering sizes.

3.2 Further Steps

A typical temporal extrapolation for an emerging minor career criminal is found in Figure 6. Here each cross represents a year of known criminal activity and the dotted line denotes the expected continuation of the sequence or criminal career.

Domain experts can easily scan such images and conclude what class of criminal career this individual will probably belong to (minor career criminal in Figure 6). If incorporation of the prediction is wanted, however, it is necessary to classify the offender under observation. Hence, we need to automatically calculate its attributes or the number of different crimes in his or her future. This can be accomplished by selecting a number of *reference points* close to the

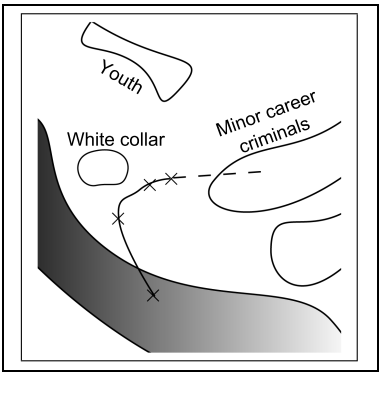


Fig. 6. Example of temporal extrapolation for criminal careers

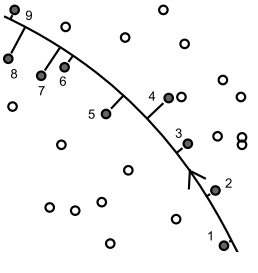


Fig. 7. Selecting points with the shortest distance to the extrapolation line

extrapolated line, averaging over their respective crime numbers to reach the expected crime data for the current offender. It was suggested in [9] that reference points closest to the last known year receive a higher weight in this process, following

$$Attrib_j(new) = \frac{2}{r+1} \cdot \sum_{i=1}^{r} (r - i + 1) Attrib_j(i),$$

where r is the amount of reference points and j is one of the crime types. This process is illustrated in Figure 7.

Of course, the number of reference points to use is a matter of accuracy versus time complexity. Looking up a large number of reference points in a database can be very time consuming, but selecting a small amount can cause accuracy to drop greatly. Selection of the right number of reference points can therefore contribute to successful implementation of this tool and is discussed in Section 4.

Now that the possible future crime numbers are calculated, the individual can easily be classified in one of the categories. Domain experts can select which categories to monitor based upon their domain knowledge, the specific needs of their own district or the specific tasks of policing they are involved in. A warning can be issued on their own computer every time a new individual falls within one of the selected danger categories.

4 Experiments

A number of experiments was performed to both reveal acceptable values for the needed parameters and test the validity of the approach as a whole. For this purpose the Dutch National Criminal Record Database was used to run the algorithm with a number of different settings. This database is available for scientific research at CBS (Statistics Netherlands) in an anonymized version. It contains approximately one million offenders and their respective crimes (approximately 50 types), of which 10% could be said to have finished their careers (no reported crimes for the last 10 years). Although this selection method is coarse, people can be incarcerated or they were simply not caught, it can still be used as a validation group.

As a first step we clustered n criminals on their (in most cases) finished criminal careers, i.e., all the crimes they committed throughout their careers. In our first test, the number of reference points was set to $r = 100$.

A ten-fold cross validation was used to calculate the accuracy of a certain method: One tenth of the population was used as a test group, where the other careers were in the clustering. All the careers in the population were "cut off" after an evenly distributed 3 or 4 years. For each of those careers, the future crime number was predicted and compared with the actual end-values of their careers. The accuracy for one career prediction will then be described by the average similarity between all 50 predicted and actual crime numbers. The accuracy of the method using these settings is then described by the mean of all averages.

Table 1. Time complexity and accuracy comparison between different methods and clustering sizes using $r = 100$ reference points

Clustering Size (n)	**1,000,000 (all)**	**100,000**	**10,000**	**1,000**
Second degree polynomial	961	98	10.4	1.0
	79.1%	77.9%	75.7%	61.3%
Third degree polynomial	965	101	10.5	1.1
	79.3%	78.2%	75.8%	62.0%
x,y **system (second degree)**	971	104	11.3	1.9
	81.5%	81.1%	79.9%	66.6%
x,y **system (third degree)**	973	105	11.7	2.1
	87.5%	87.3%	86.3%	74.4%
Spline (straight line)	982	113	**22.0**	13.4
	88.7%	88.2%	**87.3%**	74.3%
Spline (polynomial)	983	114	22.1	13.4
	79.6%	78.4%	76.9%	63.7%

For all methods, a time factor was also calculated, This time factor represents how much time was consumed, using this method within this clustering size, relative to the fastest method-size combination (which is set to 1). On standard equipment this method needed an average of 60 ms.

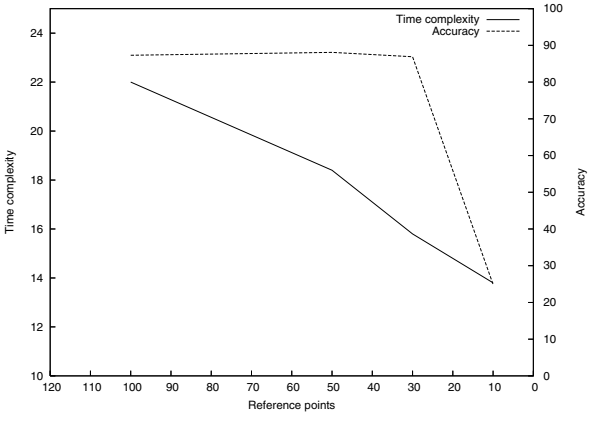
The results are presented in Table 1, where the top box of every cell describes the time factor and the bottom box contains the calculated accuracy.

From the above presented results, the conclusion can be drawn that the x,y system with a third degree polynomial and the spline with straight line extrapolation largely outperform the other methods, especially when the amount of careers in the clustering decreases. The spline performs slightly better than the system methodology.

The decrease in clustering size appears to have a minor effect on accuracy, lowering it only marginally while reducing the clustering size with a factor 100. A steep drop, however, occurs when the size is lowered to 10,000 careers. Apparently, the quality of the clustering reaches a critical low, to make a reliable prediction.

Given the time demands put on this application, the best choice would be to overlay a straight line spline extrapolation on a 10,000 size clustering (bolded option in Table 1). The accuracy of this option can be considered high and provides a solid foundation for incorporation, while allowing for fast calculation (approximately 1.3 seconds).

Potentially, an even greater gain in time can be reached by reducing the number of reference points, thus reducing calculation of the different averages. Figure 8 describes the effects on accuracy and time complexity of reference point reduction, using the optimal solution described above.



	100	**50**	**30**	**10**
Time complexity	22.0	18.4	15.8	13.8
Accuracy	87.3%	88.1%	86.9%	24.9%

Fig. 8. The relation between accuracy and time complexity when reducing the number of reference points

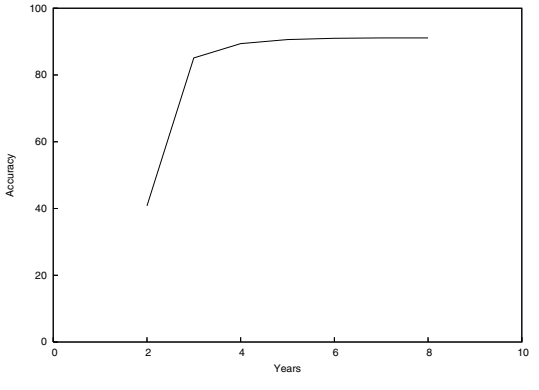


Fig. 9. Accuracy as a function of known years

Again, the reduction in information does not (necessarily) lead to decrease in quality. Reducing the number of reference points to 30, slightly lowers the accuracy with only 0.4 percentage points. Furthermore, a reduction to 50 leads to an increase of 0.8 percentage points, probably because the selection method selects careers that are simply too far away from the extrapolated line to contribute positively to the calculation of crime numbers. A steep decline can be seen with the reduction of reference points below 30. Depending on the need for speed-up or the quality of the prediction any number between approximately 50 and 30 can be selected.

It may also be of interest to see the influence of the amount of years of criminal activity that are already known on the result. In the example above, either 3 or 4 years were selected. In Figure 9, we show how the accuracy depends on the availability of realized behavior. For this experiment we used the straight line spline extrapolation, a clustering of size 10,000 and 50 reference points. Only individuals with more than 10 years of activity in total were selected for testing the extrapolation accuracy in this experiment.

As can clearly be observed in the graph, a career of which less than 3 years of activity are already recorded can not be predicted accurately. However, the results cease to improve with the addition of more than 5 years of activity. As could be expected (2 points are best extrapolated by a straight line, which in most cases does not validly predict a career), prediction efforts should start only for criminals who are active for more than 2 years, in which case an accuracy of 88% can be reached within 1.2 seconds.

5 Discussion

In this paper we demonstrated the applicability of temporal extrapolation for the extrapolation of criminal careers. This method assumes that the visualization of a clustering inherently contains a certain truth value that can yield powerful results but reduces time complexity dramatically. We superimposed such an extrapolation on an existing clustering of criminal careers and performed a series of tests that determined the speed and accuracy of the approach as well as the values of the necessary parameters. A clustering reduction was also performed to speed up calculation of a criminal career prediction.

It turns out that a clustering size of 10,000 criminal careers from the Dutch National Criminal Record Database can serve as a solid basis for extrapolation. If this extrapolation is accomplished by a spline extrapolation with straight line continuation (see [9]) and 50 reference points, accuracy can reach 88%.

As time constraints are essential to successful implementation within actual police software services, it was important to reach significant gains in computational complexity. As an end-result, all necessary tasks to be repeated for every offender to be analyzed, can be completed in approximately 1 second.

Next to the fact that predictions can immediately be visualized to police end users because of their visual nature, offender's careers can be predicted with very high accuracy in a single second. These properties make the method very well suited for incorporation in background processes at police stations, allowing alerts to be sent to dedicated machines.

A weakness to the approach, that is native to extrapolation problems, is that the lack of enough information can cause very unreliable predictions, resulting in a minimum of 3 time units of activity before anything valuable can come out of the extrapolation process. Unfortunately, data in the Netherlands is collected on a year-by-year basis, effectively establishing the demand that only third or higher year offenders can have their careers predicted.

Future research will aim at reaching even higher accuracy values by improving the selection of reference items close to the extrapolation line. A triangular shape can, among others, be employed, that selects more reference points further away (more description of the future), or closer (more reliability) from the last known data point. Also, a search for common sub-careers could be performed, that could reveal sub-careers that "define" certain classes. These subcareers, especially if they occur in the beginning of a career, could possibly improve both the speed and accuracy of future career prediction.

5.1 Reliability and Privacy

Naturally, approaches such as this one come with some reliability and privacy concerns. Most data mining applications concern quests for truth or the discovery of general knowledge, rather than the translation of such newly discovered patterns or descriptions to individual cases. This approach is an exception to this situation, which raises some questions about the statistical validity of the results when applied in real life situations, especially areas as important as that of law enforcement.

This approach is best compared to the process of discovering suspicious financial transactions like for example in [12]. These methods also enforce criteria from a knowledge discovery support system on individual financial transactions in order to find outliers that might indicate for example money laundry activities. Naturally, not all transactions that yield warnings are actually fraudulent, but as long as the expected chance a warning is actually correct is reasonably high, the usage of such an decision support system is warranted as long as each warning is reviewed by experts before action is undertaken. The same should apply to the usage of our approach. Still it remains an important warning to police officers working with the software that for every 88 careers the system predicts correctly, there are 12 careers predicted incorrectly with varying margins.

Another important concern is that of privacy. Constant electronic monitoring of one's activities can be seen as an invasion on one's personal privacy as stressed in, for example, [15]. However, in contrast with, among others, the above mentioned financial monitoring systems, that monitor *all* transactions done by *everybody*, only the people who have been active criminals for more than three years are under surveillance and even then they are only electronically processed every time they commit a new crime. Therefore, the approach as described in this paper poses a minimal threat to the privacy experience of the regular population.

Acknowledgment

This research is part of the DALE (Data Assistance for Law Enforcement) project as financed in the ToKeN program from the Netherlands Organization for Scientific Research (NWO) under grant number 634.000.430.

References

1. Abdi, H.: Signal detection theory. In: Salkind, N.J. (ed.) Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks (2007)
2. Adderley, R., Musgrove, P.B.: Data mining case study: Modeling the behavior of offenders who commit serious sexual assaults. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001), New York, pp. 215–220 (2001)
3. Blumstein, A., Cohen, J., Roth, J.A., Visher, C.A.: Criminal Careers and "Career Criminals". National Academies Press, Washington (1986)
4. Chau, M., Atabakhsh, H., Zeng, D., Chen, H.: Building an infrastructure for law enforcement information sharing and collaboration: Design issues and challenges. In: Proceedings of The National Conference on Digital Government Research (2001)
5. Chau, M., Xu, J., Chen, H.: Extracting meaningful entities from police narrative reports. In: Proceedings of The National Conference on Digital Government Research, pp. 1–5 (2002)
6. Chen, H., Atabakhsh, H., Petersen, T., Schroeder, J., Buetow, T., Chaboya, L., O'Toole, C., Chau, M., Cushna, T., Casey, D., Huang, Z.: COPLINK: Visualization for crime analysis. In: Proceedings of The National Conference on Digital Government Research, pp. 1–6 (2003)
7. Cocx, T.K., Kosters, W.A.: A distance measure for determining similarity between criminal investigations. In: Perner, P. (ed.) ICDM 2006. LNCS (LNAI), vol. 4065, pp. 511–525. Springer, Heidelberg (2006)
8. Cocx, T.K., Kosters, W.A., Laros, J.F.J.: Enhancing the automated analysis of criminal careers. In: SIAM Workshop on Link Analysis, Counterterrorism, and Security 2008 (LACTS 2008) (2008)
9. Cocx, T.K., Kosters, W.A., Laros, J.F.J.: Temporal extrapolation within a static clustering. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) Foundations of Intelligent Systems. LNCS (LNAI), vol. 4994, pp. 189–195. Springer, Heidelberg (2008)
10. Davison, M.L.: Multidimensional Scaling. John Wiley and Sons, New York (1983)
11. The FBI strategic plan (2004–2009), http://www.fbi.gov/
12. Goldberg, H.G., Wong, R.W.H.: Restructuring transactional data for link analysis in the FinCEN AI system. In: Papers from the AAAI Fall Symposium, pp. 38–46 (1998)
13. Kosters, W.A., Laros, J.F.J.: Metrics for mining multisets. In: Proceedings of the Twenty-seventh SGAI International Conference on Artificial Intelligence SGAI 2007, pp. 293–303 (2007)
14. Oatley, G.C., Zeleznikow, J., Ewart, B.W.: Matching and predicting crimes. In: Proceedings of the Twenty-fourth SGAI International Conference on Knowledge Based Systems and Applications of Artificial Intelligence (SGAI 2004), pp. 19–32 (2004)
15. Schermer, B.: Software Agents, Surveillance, and the Right to Privacy: A Legislative Framework for Agent-enabled Surveillance. PhD thesis, Leiden University (2007)
16. Xiang, Y., Chau, M., Atabakhsh, H., Chen, H.: Visualizing criminal relationships: Comparison of a hyperbolic tree and a hierarchical list. Decision Support Systems 41(1), 69–83 (2005)

Comparing Distance Measures with Visual Methods

Isis Bonet[1], Abdel Rodríguez[1], Ricardo Grau[1], Maria M. García[1], Yvan Saez[2], and Ann Nowé[3]

[1] Center of Studies on Informatics, Central University of Las Villas, Santa Clara, Villa Clara, Cuba
[2] Department of Plant Systems Biology, Flanders Interuniversity Institute for Biotechnology (VIB), Ghent University, Belgium
[3] Computational Modeling Lab, Brussels University, Belgium
isisb@uclv.edu.cu, abdelr@uclv.edu.cu, rgrau@uclv.edu.cu, mmgarcia@uclv.edu.cu, yvsae@psb.ugent.be, ann.nowe@como.vub.ac.be

Abstract. The selection of the distance measure to separate the objects of the knowledge space is critical in many classification algorithms. In this paper, we analyze the distance measures reported in the literature for the problem of HIV prediction. We propose a new distance for HIV viral sequences, based on the mutations with regard to the HXB2 reference sequence. In a first step, we reduce data dimensionality in order to subsequently analyze the distance measure's performance in terms of its ability to separate classes.

Keywords: MDS, kNN, similarity, distance measure, HIV.

1 Introduction

The selection of an appropriate distance function to separate the objects of a given problem is an important and non-trivial task in both supervised and unsupervised pattern recognition techniques [1].

In supervised learning, a class label is assigned to all instances and the goal is to find an appropriate classification model, capable to learn from the data and assign the right class to an unseen instance. We can divide this process in the following steps: preprocessing of the data, training of the classifier, and evaluation (testing).

On the other hand, in an unsupervised learning problem, the task is to group unlabeled instances in clusters without a priori knowledge of the classes.

Both supervised and unsupervised algorithms follow a similar setup, where the first task consists of a preprocessing step. Preprocessing involves feature extraction and selection, and is necessary in a wide range of problems. In the extraction step, the use of a similarity function is quite common, and well known unsupervised dimensionality reductions such as Principal Component Analysis (PCA) [2], Sammon´s mapping, and Multidimensional Scaling (MDS) [3], are all based on distance matrices.

There are many clustering methods that need the definition of this distance function, k-means [4] for example, or even artificial neural networks like

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 90–99, 2008.

self-organizing map (SOM) [5], but there also exist supervised models such as instance-based learning algorithms like the k-nearest neighbor algorithm (kNN) [6], or kernel methods [7].

Both methods, supervised and unsupervised, are useful in wide extended areas in science. Bioinformatics is an important area where machine learning techniques are broadly used nowadays to solve an extensively kind of problems. RNA, DNA and protein sequence analysis are clear examples of these applications.

Alignments and sequence similarity scores are commonly used to compare amino acids sequences when we are facing evolutionary history inferences, or protein families' biological function characterizations. There are different scoring schemes. One of them are scoring matrices from PAM [8] and BLOSUM [9] families, arising in particular to emphasize evolutionary and biological properties. But most of classic methods, non-parametrical statistics and some artificial neural networks are still based on instances distance measure.

Some authors have obtained their distance functions based on these score matrices. Others applied chain distances, taking the biological amino acids sequence as a chain. In spite of the wide variety of measures reported in the literature, researchers are still looking for new functions to separate in a better way the patterns of specific problems [10, 11].

When we face a pattern recognition problem, as we said before, no matter it is supervised or unsupervised, preprocessing is a first step in analyzing the data. Ideally, we must completely know the data we are working with, and visually representing the data can be an important help in order to study its characteristics.

In this paper we present a novel distance measure for HIV sequences. We show how to support the search for an appropriate distance using a visual representation of the dataset after having applied dimensionality reduction techniques.

2 Methods

2.1 Problem Formulation

For the HIV resistance prediction problem, we use the Stanford HIV Resistance Database [12]. This dataset is described by a genotype-phenotype relation and describes some protease inhibitors. We select 6 well known protease protein inhibitors: amprenavir (APV), atazanavir (ATV), indinavir (IDV), nelfinavir (NFV), ritonavir (RTV), and saquinavir (SQV) for this work. This data are described by the genotype and the phenotype.

The genotype is described by the list of mutations with respect to a reference sequence. We modeled each genotype sample using one attribute for each amino acid. Each amino acid was replaced by its respective contact energy, due to the relation between this property and the 3D structure, resulting in continuous valued features [13].

The phenotype is described by the resistance-fold based on the concentration of the drug to inhibit the viral protease. This value is used to turn the dataset into a classical classification problem, taking as cut-off the standard value of 3.5, as described in the literature. Thus, if the resistance-fold is larger than the cut-off value, the viral

sequence is regarded as resistant to the drug; else the sequence is termed to be susceptible to the drug. Finally, we now have 6 classification problems with 99 descriptive features and only one objective feature.

2.2 Distance Measures

As we mentioned before, there are a lot of distance measures already defined, used in a really sparse kind of problems. Some of them are metrics, also called distance functions. Some others are semi-metrics or pseudo-metrics. The most well known distance function is Euclidean distance, but it is not applicable to all problems. Other distance functions include Mahalanobis, Minkowski and correlation based functions. The Value Difference Metric (VDM) is defined over nominal attributes and there even exist metrics developed for a mixture of nominal and continuous attributes [1].

In order to compare the results of our new distance measure, we will be using some well known functions. Let us describe these functions taking into account that X and Y are the objects we will compare, with dimension N (attribute number), and x_i and y_i denote the i^{th} attributes of X and Y respectively.

Euclidean distance (equation 1) is a metric for continuous values as we already said. Taking into account that in our HIV resistance prediction problem we use contact energies to represent each amino acid in the sequence, this is a continuous value, and then this function is applicable.

$$D(X,Y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{1}$$

Manhattan distance (also known as City Block distance) is also defined over continuous attributes. It represents the total energy variation in mutated positions (equation 2).

$$D(X,Y) = \sum_{i=1}^{N} |x_i - y_i| \tag{2}$$

Chebychev distance (equation 3) represents the maximum energy variation in our problem.

$$D(X,Y) = \max_{i=1}^{N} |x_i - y_i| \tag{3}$$

Hamming distance represents the total different positions between two sequences. It measures the minimum substitution number needed to transform a sequence into the other (equation 4).

$$\begin{aligned} D(X,Y) &= \sum_{i=1}^{N} d(x_i, y_i) \\ d(x_i, y_i) &= \begin{cases} 1 & si \quad x_i \neq y_i \\ 0 & si \quad x_i = y_i \end{cases} \end{aligned} \tag{4}$$

Levenshtein distance measures the minimum operations number required to transform a sequence into the other. By operation we mean insertion, deletion, or substitution of amino acids. This is a generalization of *Hamming distance* not taking just substitution into account, but insertions and deletions too.

In the HIV protease drug resistance dataset all sequences have the same length, it has no insertion or deletion, so the *Levenshtein distance* is exactly equal to *Hamming distance* and we will not use it in our tests. A final note concerns the *Hamming distance* which counts the different positions in the sequence. This means that the sequence is treated as a chain of nominal values. It is effective to our problem, because each of the 20 possible amino acids has a unique energy, so this energy could be used as a nominal value as well.

2.3 Mutations Based Distance Measure

The *Hamming* and *Levenshtein* distances could be seen as measures that take into account the mutations in the sequence. There are some other distances with the same assumption [14]. There are also distances or dissimilarity measures based on the probability of substitution of each amino acid by another stored in score matrices.

In this paper we discuss a new distance based on the mutations number taking HXB2 (wild-type sequence) as reference pattern for the HIV sequences.

Let $R = (r_1,r_2,...r_N)$ be the HXB2 reference pattern, and $X = (x_1,x_2,...,x_N)$ and $Y = (y_1,y_2,...,y_N)$ the sequences to be compared; the dissimilarity measure is defined in equation 5, where $S(X,Y)$ is the similarity measure defined in equation 6.

$$D(X,Y) = 1 - S(X,Y) \tag{5}$$

$$S(X,Y) = \frac{\sum_{i=1}^{N} similarity(x_i, y_i)}{|M(X) \cup M(Y)|} \tag{6}$$

Where $M(X)$ denotes the mutated attributes set in sequence X taking R as reference pattern. The denominator in equation 6 represent the cardinality of the union of mutated attributes sets, that is the total number of attributes that have mutated in X or Y. The $similarity(x_i,y_i)$ represents the similarity of features, and is defined in equation 7.

Note there are different score w_i for different described situations, in order to emphasize the differences between the mutations. It is highly recommended that $w_1 > w_2 > w_3$, so the highest score is for the attribute with the same mutation in both sequences (w_1); there is an intermediate score for attributes with different mutations but in the same direction of energy change (w_2); and finally a the lowest score is awarded to the rest of attributes that have mutated in both sequences but have no similarity at all (w_3). There is no reward for attributes where there is only a mutation in one of the sequences.

$$similarity(x_i, y_i) = \begin{cases} 0 & if\ (x_i = r_i) \vee (y_i = r_i) \\ w_1 & if\ (x_i \neq r_i) \wedge (y_i \neq r_i) \wedge (x_i = y_i) \\ w_2 & if\ (x_i \neq r_i) \wedge (y_i \neq r_i) \wedge (x_i \neq y_i) \\ & \quad \wedge \operatorname{sgn}(r_i - x_i) = \operatorname{sgn}(r_i - y_i) \\ w_3 & if\ (x_i \neq r_i) \wedge (y_i \neq r_i) \wedge (x_i \neq y_i) \\ & \quad \wedge \operatorname{sgn}(r_i - x_i) \neq \operatorname{sgn}(r_i - y_i) \end{cases} \quad (7)$$

Keep in mind $sgn(X)$ is the sign function.

2.4 Dimensionality Reduction

There are several techniques to reduce dimensionality as we mentioned before. We have chosen MDS due to the distance matrices manipulation advantage. The final goal is to visualize the data transformed using the different distances in order to select the one that is best capable of separating them, as a potential effective distance to be used in distance based methods. Here, we chose the kNN classifier to test our methodology.

3 Experimental Results

Let us show the results for the process predicting the resistance of the HIV protease with 6 drugs adding the new mutations based distance.

First of all, each drug is reduced to two dimensions and it is visualized in order to look for visual information about the consequences of using the distance to separate the classes.

Keep in mind the considerations made above about the appropriate w_1, w_2 and w_3 for the last distance. We select the maximum value when there is the same mutation in both sequences (w_1=1). We reduce the reward to a half when there are different mutations in the same direction, both increase or decrease the energy (w_2=0.5). Finally we also reduce the reward to a half when there are mutations but neither in the same direction (w_3=0.25).

These datasets are described by 99 amino acids, so the task of reducing this amount of features to only 2 is really complex. We are facing here a really hard problem in terms of separation of classes, and it is easily seen in figures 1, 2 and 3. In spite of this, it can be easily seen which measure obtains the best separation of classes.

Figure 1 shows the results of the dimensionality reduction of IDV and NFV using Euclidean, City Block, Hamming, Chebychev and our new distance (*WeightedMutation*). Each point corresponds to a case of the database; black points represent a class whereas gray points were used for the other class. In both drugs, our novel distance clearly performs in a better way.

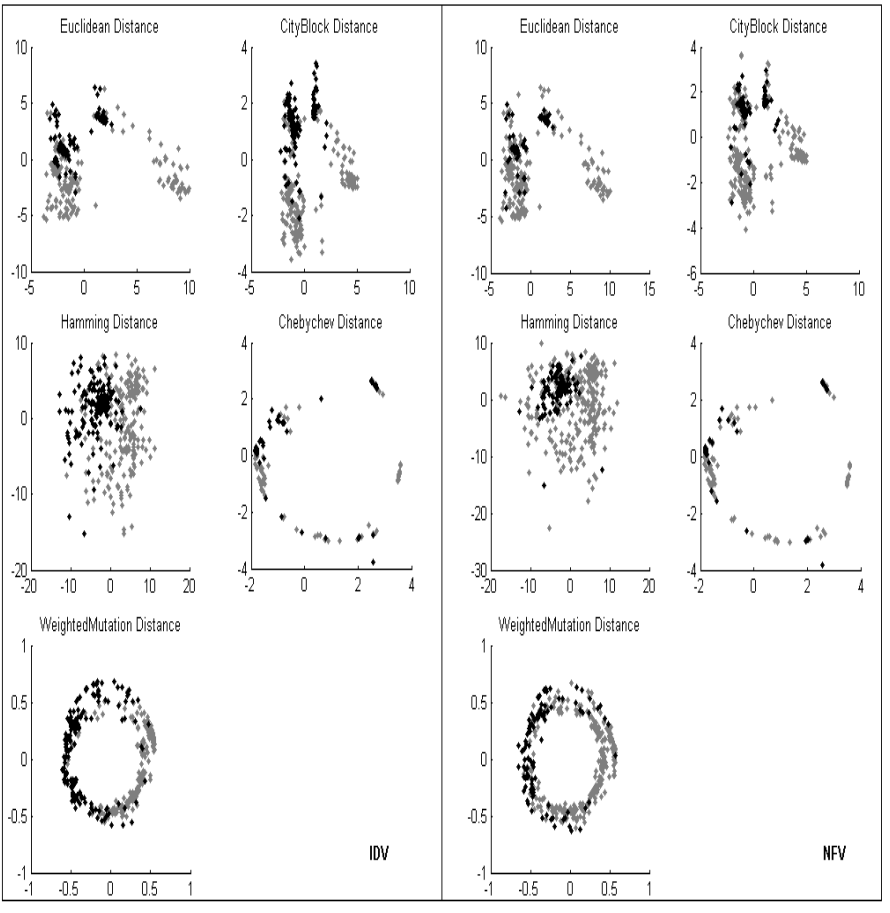


Fig. 1. IDV and NFV datasets reduced by MDS using Euclidean, City Block, Hamming, Chebychev, WeightedMutation distances

We can see the *WeightedMutation* distance draws a circle with the cases, clearly separating objects from different classes in a better way than the other distances. If a vertical line is drawn near to the center of the circle in NFV, nearly all of the black points (one class) lie at left side and most of the gray points (the other class) at the other side. In the representation of the cases using the remaining distances is harder to divide the classes.

Figure 2 depicts the results of MDS for APV and RTV. Here is also possible to see that the separation between the classes using *WeigtedMutation* distance is easier to be obtained than with the other distances.

On the other hand, the same does not occur for ATV and SQV. It can be seen in figure 3 that the 2D representation for these drugs does not really offer useful

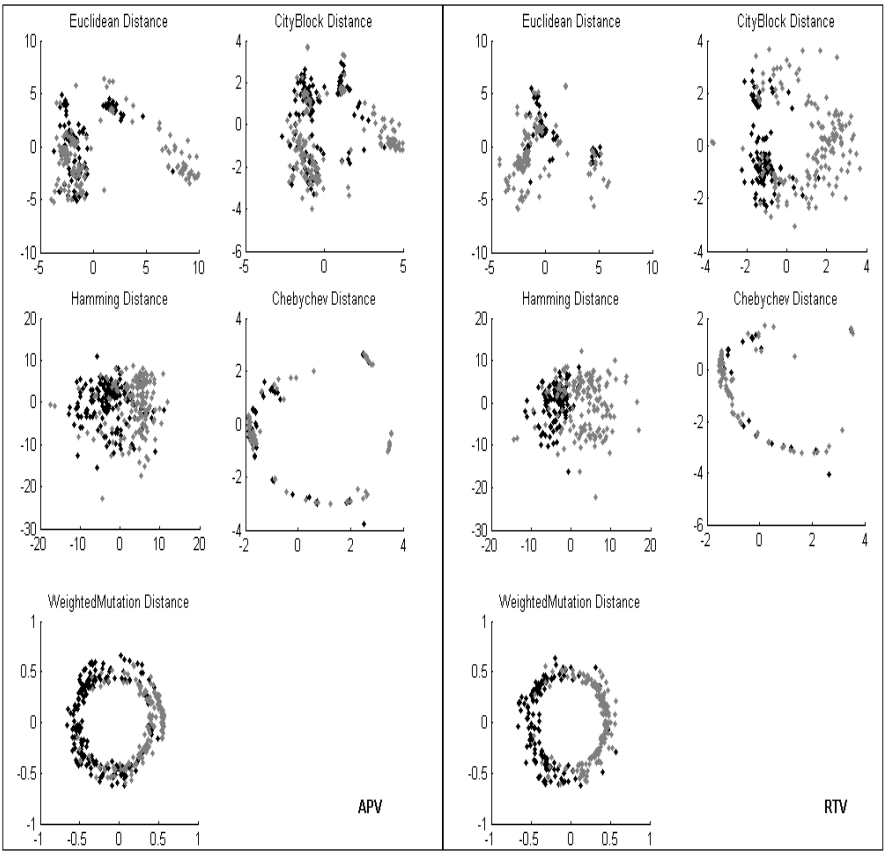


Fig. 2. APV and RTV reduced by MDS using Euclidean, City Block, Hamming, Chebychev and *WeightedMutation*

information about which distance is the best because the points of two classes are merged.

Once the visualization has been analyzed, the remaining experiment is to verify these results using a distance based classification method with the original databases with 99 features. We choose the kNN classifier as the classification method using the same distance measures that were used for the visualization. We pick k=15 and use 10-fold cross-validation method to evaluate the distance measures.

Up to now our hypothesis is the best result should be obtained using *WeightedMutation* distance for APV, IDV, NFV and RTV, but not for ATV and SQV.

Figure 4 shows the kNN performance for each drug. The results obtained here support our hypothesis. *WeightedMutation* distance is the winner in APV, IDV, NFV and RTV, which is just what we were expecting. The same does not occur in SQV

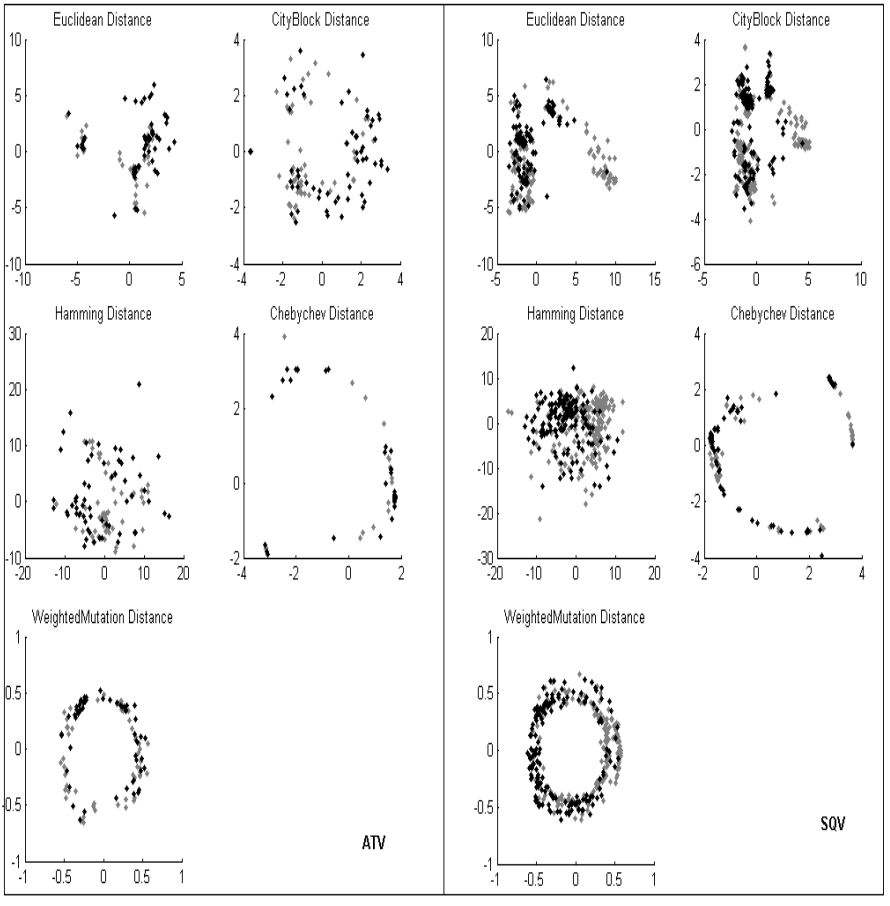


Fig. 3. ATV and SQV datasets reduced by MDS using Euclidean, City Block, Hamming, Chebychev, WeightedMutation distances

and ATV. In fact, the results reached by kNN with any of these distances are low for ATV and SQV, as we expected.

In order to analyze the stability of the results the standard deviation was computed. Our distance achieves 0.045 as average for APV, IDV, NFV and RTV whereas the remaining distance averages range from 0.060 to 0.066.

All in all, when we need to use a distance measure in a supervised problem is useful to utilize visual information before going for the appropriate one. Dimensionality reduction can be firstly done using a method based on a distance measure as MDS. A visual analysis can then be performed in order to decide which of the possible measures can achieve a better separation between the classes.

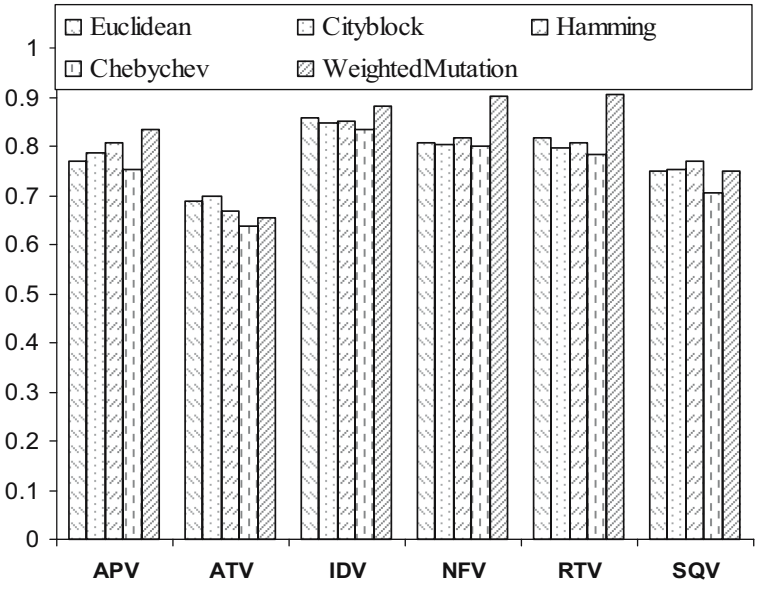


Fig. 4. kNN accuracy results using Euclidean, City Block, Hamming and Chebychev distances for APV, ATV, IDV, NFV, RTV and SQV datasets

4 Conclusions

The results obtained in this paper show that the new proposed distance based on weighted mutations is a promising distance measure for the problem of HIV resistance prediction. We showed that the use of dimensionality reduction techniques such as MDS help us to select the appropriate distance for a given problem to be used in future classification.

References

1. Wilson, D.R., Martinez, T.R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)
2. Jolliffe, I.T.: Principal Component Analysis. Springer, New York (1986)
3. Cox, T., Cox, M.: Multidimensional Scaling. Chapman and Hall, Boca Raton (1994)
4. McQueen, J.: Some methods for classification and analysis of multivariate observations. In: Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 182–297 (1967)
5. Kohonen, T.: Self-Organization and Associative Memory. Springer, New York (1989)
6. Cover, P.E., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
7. Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12(2), 181–201 (2001)
8. Dayhoff, M.O., Schwartz, R., Orcutt, B.C.: A model of evolutionary change in proteins. Atlas of Protein Sequence and Structure 5, 345–352 (1978)

9. Henikoff, S., Henikoff, J.: Amino acid substitution matrices from protein blocks. Proccedings of the National Academy of Sciences of the United States of America. PNAS 89, 10915–10919 (1992)
10. Baldi, P., Soren, B.: Bioinformatics: The Machine Learning Approach. MIT Press, Cambridge (2001)
11. Bhaskar, H., Hoyle, D.C., Singh, S.: Machine learning in bioinformatics: A brief survey and recommendations for practitioners. Computers in Biology and Medicine 36(10), 1104–1125 (2006)
12. Stanford HIV Resistance Database Protease, `http://hivdb.stanford.edu//cgi-bin/GenoPhenoDS.cgi`
13. Miyazawa, S., Jernigan, R.L.: Residue Potentials with a Favorable Contact Pair Term and an Unfavorable High Packing Density Term, for Simulation and Threading. J. Mol. Biol. 256, 623–644 (1996)
14. James, R.: Predicting Human Immunodeficiency Virus Type 1 Drug Resistance from Genotype Using Machine Learning. University of Edinburgh (2004)

Semantic Annotation and Question Answering of Statistical Graphs

Michel Dumontier[1,2], Leo Ferres[3], and Natalia Villanueva-Rosales[2]

[1] Department of Biology
[2] School of Computer Science
Carleton University, Ottawa, Canada
[3] Department of Computer Science
University of Concepción, Concepción, Chile
michel_dumontier@carleton.ca,
leo@inf.udec.cl,
nvillanu@scs.carleton.ca

Abstract. Although statistical graphs are ubiquitous, few techniques and standards exist to exchange, search and query these graphical representations. The gSem project aims to improve human-graph interaction by developing new approaches to manage statistical graph knowledge. Two specific objectives of this project are addressed in this paper: (1) improve the efficiency of searching statistical graph knowledge and (2) facilitate sophisticated question answering about statistical graph knowledge. In particular, semantic annotation and query answering across statistical graphs with OWL ontologies are described.

Keywords: Ontology, question answering, OWL, statistical graphs, semantic web.

1 Introduction

Statistical graphs (bar and line graphs, pie charts, stacked bar graphs, etc. such as that in Figure 1) are visual objects aimed at either communicating or exploring numerical data. Although ubiquitous, there exist few techniques and standards in place to exchange, search and query these graphical representations. Nowadays, the most common publication format for statistical graphs is using image technologies that produce raster images (e.g. gif, jpg, etc). These unstructured formats are enough to communicate their content and meaning in certain situations (e.g. during a business meeting, embedded in a report and described with natural language, etc.). However, the representation of statistical graphs in an unstructured format has several important drawbacks, such as i) the inability to exchange or merge complementary graphs, ii) the inability to retrieve graphs having particular content that lies within a certain value range, and iii) the inability to answer questions that require deeper knowledge of a graph's general purpose. The gSem project aims to improve human-graph interaction, overcoming these deficiencies with new approaches to manage statistical graph knowledge. This

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 100–110, 2008.

Fig. 1. A graph appearing on "The Daily" website on Monday, July 18, 2008

project has the following objectives: (1) improve the efficiency of searching statistical graph knowledge, (2) facilitate sophisticated question answering about statistical graph knowledge, (3) increase the accessibility of statistical graphs for the visually impaired, and (4) enable graph re-purposing beyond their original communicative intent.

In creating graphs for mass consumption, authors usually have some communicative purpose in mind [1,2]. Thus, while graph data may be structured and exchanged using a common exchange format such as OpenXML (commonly referred to as OOXML or Open XML) [3] or Open Document Format [4], important information about the semantics of graphs may still be lost. For instance, an author may beed to create two graphs from collected data: one for an investor, in which the author shows the increase of sales over a period of five years, and one for management, in which he or she breaks down the sales into the sale of goods and the sale of services per quarter. With XML it would be challenging to know that both Goods Sales and Service Sales are types of Sales and that these are not the same. As well, XML data is hierarchically organized with arbitrary relationships between elements, in which case the *semantics* of the relationship (i.e. **isA**, **partOf**) is either implicit or unknown. Finally, the author of the graph may have put special effort in communicating certain aspects of the data such as choosing one scale over another, annotating one part of the graph with a different color, etc. The addition of this information requires a semi-structured approach in which other groups can easily add new knowledge to the graph.

In previous work [5], we extended an XML representation of statistical graphs [6] with a more formal knowledge representation approach: a statistical graph ontology using the OWL Semantic Web language. One variant, OWL-DL, implements a subset of Description Logics, a formalism characterized by i) the use of

different kinds of constructors to define complex concepts out of simpler ones and ii) an emphasis on the soundness, completeness and computational tractability of reasoning problems [7]. OWL-DL offers many salient features for knowledge representation including subsumption, class constructs (i.e. union, intersection, negation), class equivalence for automatic classification, property characteristics (transitive, inverse, symmetric, functional, inverse functional), qualified cardinality restrictions and the explicit assertion of identity or difference between entities.

In this work, we aim to (a) provide a methodology that allows semantic annotation of graphs and bridges the gap between Web 2.0 social tagging and Semantic Web ontologies (Web 3.0), (b) demonstrate knowledge discovery across a graph knowledge base composed of an instantiated ontology and (c) demonstrate sophisticated question answering that requires background knowledge. Novel aspects of our work include: (i) a clean methodology for the representation and semantic annotation of statistical graphs, (ii) knowledge management of statistical graphs based on semantic web technologies, (iii) information retrieval across graphs with semantic overlap, (iv) sophisticated question answering that include negated queries based on ontological knowledge. This work forms an important milestone towards the creation of semantically annotated graphs that are machine understandable, compatible with semantic web technologies, and can be queried with precise semantics.

2 Knowledge Representation of Statistical Graphs

Communicative statistical graphs (CSGs) are normally *visually* simple: they usually contain only a few number of points (in our corpus, less than 200), only one or two horizontal (category) axes and generally no more than three or four series. Figure 1 shows a sample communicative graph. The translation from visual to propositional, however, has always been a challenging task [8]. In the case of (short) time-series data such as the ones we are dealing with in this paper, several different dimensions can be semantically encoded. For instance, we could import time ontologies [9,10] to reason with value points as being related by time relations such as **isBefore**; we could import spatial ontologies [11,12] to reason with value points as **isAbove** or **isAdjacentTo** certain other graphical objects, to name but two of many possible complementary annotations.

The SGO ontology proposed in [5] is extended in this work. Briefly, a graph is defined as follows: A Graph **hasPart** one or more Plot and may (**hasTitle**) one or more Title (PrimaryTitle or SecondaryTitle). The Plot **hasPart** one or more Series and **hasPart** one or more CoordinateAxis (XAxis, YAxis or ZAxis) which may **hasPart** one or more CategoryAxis (PrimaryCategoryAxis or SecondaryCategoryAxis) or one or more ValueAxis (LeftValueAxis or RightValueAxis).

A CategoryAxis **hasPart** CategoryData, and more specifically, a PrimaryCategoryAxis **hasPart** PrimaryCategoryData whereas SecondaryCategoryAxis **hasPart** SecondaryCategoryData. CategoryData may be ordered using **isBefore** or **isAfter** properties. In contrast, a ValueAxis may be used to plot continuously varying numerical data (ValueData) and may be scaled (ScaledValueAxis such as LinearValueAxis, LogarithmicValueAxis and SemiLogarithmicValueAxis).

A Series **hasPart** two or more DataPoint **hasPart** Data to be plotted from the set of CoordinateAxis. Data, or more specifically CategoryData or ValueData **hasValue** a concrete datatype value (e.g. int, float, double). Certain properties are defined as transitive (**hasPart** and its inverse **isPartOf**) such that if *graph1* **hasPart** *plot1* and *plot1* **hasPart** *series1*, then a *graph1* **hasPart** *series1*.

3 Graph Processing

The iGRAPH-Lite framework is used in this work. It contains application specific plug-ins (MS Excel, SPSS, OpenCalc, etc) to translate the internal representation of a graph into a Visual Description V_{desc} of that graph [13]. The plug-ins extract a minimal and common set of information, which is only part of the objects and their relations needed for the OWL representation. Once the V_{desc} has been generated, a set of algorithms are identified from the *concept repository*, which describes the name and location of algorithms to be loaded by the system. These algorithms identify or enrich content that can potentially be communicated about the graph. For instance, one algorithm determines whether the slope between any two points in the graph is increasing or decreasing. Another simple MATCH algorithm checks for the type of graph (a line graph, in this case). Each algorithm generates additional information which is added to the enhanced XML representation (iGraphXML).

The iGraphXML document is subsequently transformed by an XML stylesheet[1] as instances of the SGO ontology. The XML stylesheet generates the semantic annotation (described in next section) for each one of the category axis and value axis instances using syntactic matching of the iGraphXML file annotations and the predefined keywords covering expected time intervals and value axes (e.g. "Year", "Month", "Quarter", "Billion", etc.). For illustration purposes, an example knowledge base[2] containing the graph ontology and four graphs[3] was generated in this fashion. These examples, whose visual representations can be seen in Figure 2, are representative of a much larger and daily increasing knowledge base that currently contains more than 200 statisticals graphs from "The Daily"[4].

4 Semantic Annotation

An interesting aspect of our work focuses on the semantic annotation of graphs to further augment the knowledge about the graph components. For instance, we might want to say that each of the value data in *graph1* corresponds to "service

[1] http://ontology.dumontierlab.com/igraph-mapping.xsl

[2] http://ontology.dumontierlab.com/graph-example-all.owl

[3] Individual graphs are available at http://ontology.dumontierlab.com/graph-example-X.owl, where X is the graph number (1-4).

[4] http://www.statcan.ca/english/dai-quo/

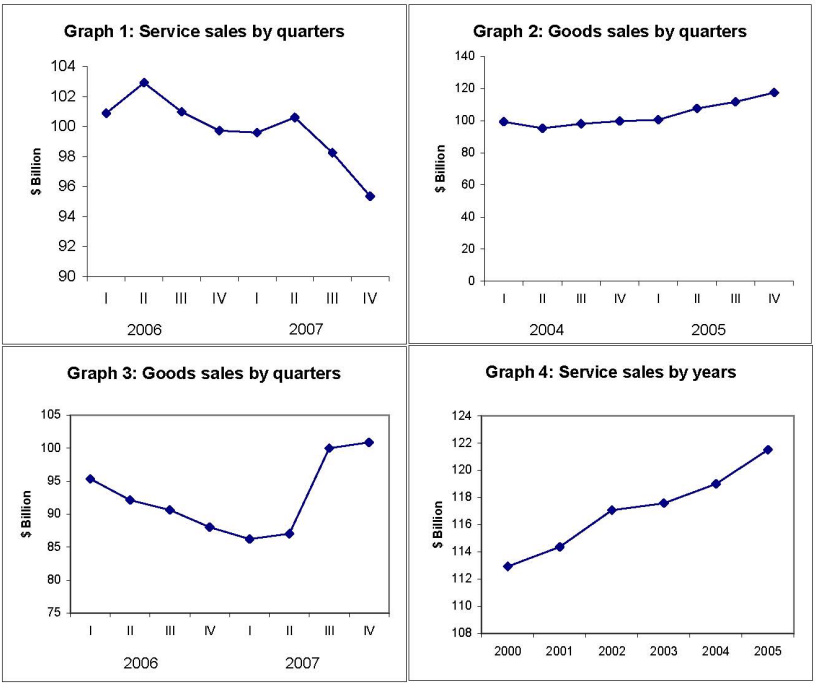


Fig. 2. Visual representations of the example graph knowledge base that were represented using the statistical graph ontology

sales", or that each of the secondary category data in *graph1* corresponds to specific years (i.e. the year 2006 or the year 2007). Since social tagging is popular, distributed and scalable, there exists an important requirement to bridge the gap between formal and informal knowledge representations. For this purpose, we have introduced a set of object properties (**hasTag**/**isTagOf**) that can be used with minimal ontological commitment. In the weakest case, graph components can be related to a generic instance of some unknown category, being identified by some Uniform Resource Identifier (URI) and having some human readable label (`rdfs:label`). A stronger commitment can be made by making this an instance (`rdf:type`) of some class defined in some RDF/OWL document, thus providing the ability to reason about its class membership. This object property approach is therefore more flexible than the alternative of datatype annotation alone, which could only be subject to syntactic matching, not taking advantage of reasoning over existing ontologies. Another valuable aspect of this approach is that we can discover graphs having components that are semantically annotated. To infer that graphs having semantically annotated parts are themselves semantically annotated, we use the object property chain inclusion axiom [14] [**hasPart** ∘ **hasTag** → **hasTag**], asserting that the composite property **hasPart** ∘ **hasTag** is a subproperty of **hasTag**. In this way, graphs can be queried directly

with the tags (see use cases below for worked examples). Semantic annotations based on keywords or controlled vocabularies can later be mapped to ontologies for enhanced question answering about graph contents (e.g. time intervals, (currency) units, sales)[5].

5 Data Integration

Data integration of these OWL-based statistical graphs occurs at the semantic level, rather than at a purely syntactic or identity level. The identity of each graph component (i.e. `http://ontology.dumontierlab.com/graph-example-1.owl#x1`) is based on a sufficiently unique base URI (i.e. `http://ontology.dumontierlab.com/graph-example-1.owl`) which could be generated from a unique number server or by some hash of the contents such that no collisions will occur in a large knowledge base. Data integration is achieved by the semantic annotation by type. For instance, all graphs will be instances of (Graph) and all y-axis value data will be instances of (ValueAxis and **isPartOf** YAxis). Similarly, data that is semantically annotated with instances of classes defined in other ontologies (`sales:ServiceSales`) can be integrated at the class level. The only entities that are consistently reused are the specific instances of required time intervals (i.e. quarter 1 of 2006, year 2006). This has the distinct advantage of being able to query for anything in the knowledge base that is annotated with named time intervals and disambiguation of these time intervals with respect with other units (e.g. 1996 is a year and not a dollar amount).

6 Question Answering

Statistical graphs represented using our approach have the distinct advantage of being semantically queried such that they can answer questions that require a background knowledge. This background knowledge is encoded in the OWL ontologies that are reused via semantic annotations. Here, we explore three use cases to illustrate queries that i) integrate graphs, ii) differentiate graphs, iii) span graphs. In order to execute these queries, the user loads the example knowledge base into Protege 4 and reasons with either the Pellet or FaCT++ reasoner. The user formulates a query using the Manchester OWL syntax [15] via the Protege 4 DL Query tab.

Use Case 1: Identify Relevant Time-Series Graphs. To identify all service sales graphs between 2003 and 2006, Jill formulates the following query:

```
Graph that
  (hasTag some ServiceSales) and
  (hasTag some
    (Year and
```

[5] All ontologies available at `http://ontology.dumontierlab.com`

```
    isAfter some {_2002} and
    isBefore some {_2007}))
```

This query returns two graphs: *graph1* and *graph4*, see Figure 2. This answer stems from the following features: i) a **hasTag** role chain - this rule infers that for every x that has a part y, in which y has tag z, then x has tag z - and this leads to two inferences: a) since the graph has a series as a part, and that series is annotated as a service sales, then the graph is also annotated with services sales, and b) since each graph has categorical data annotated to specific years, then the graph is also annotated with these time intervals; ii) the imported instantiated time-interval ontology contains information that asserts the relative ordering of years and quarters using **isImmediatelyBefore** (asserted) and its inverse **isImmediatelyAfter** (inferred), which are sub-properties of the transitive **isBefore**/**isAfter** property, respectively.

Alternatively, Jill may wish to identify all goods sales that are not included in *graph2*:

```
Graph that
  (hasTag some GoodsSales) and
  (hasTag some
    (Year and
    isBefore some {_2004} or
    isAfter some {_2005}))
```

This retrieves *graph3* with information about 2006 and 2007.

Use Case 2: Differentiating semantically annotated graphs to identify all graphs with sales, excluding service sales, Joe creates the following query:

```
Graph that
  (hasTag some (Sales and not ServiceSales))
```

This query returns two graphs: *graph2* and *graph3*. This answer comes from two main features: i) the **hasTag** role chain, and ii) the knowledge that Goods Sales is not a type of Service Sales is obtained from the Sales ontology which asserts that Goods Sales and Service Sales are disjoint and that they are both types of Sales.

Use Case 3: Searching iGraph Augmented Graphs, Jane would like to determine which quarters between 2006 and 2007 have increasing sales.

```
Quarter that
  isPartOf some ({_2006} or {_2007}) and
  isTagOf some
    (isStartPointOf some IncreasingLine and
    isPartOf some
      (Series that hasTag some Sales))
```

This query returns the four quarters that meet this criteria: *I_2006*, *I_2007*, *II_2007*, *III_2007*. Answering this question was made possible due to several aspects: i) the instantiated time interval ontology associates four uniquely named quarters for each of the two years in question using a *part of* relation, ii) we discover that these quarters are inversely related to the x-axis category data by the **hasTag** property, iii) these category data are part of a data point that is the starting part of an increasing line identified by iGraph, iv) the category data is part of a series that has been tagged either as Goods Sales or Service Sales, which are in turn subsumed by Sales. The four quarters were identified from two graphs (*graph1*, *graph3*).

7 Discussion

For simplicity, an example knowledge base was used to illustrate the advantages of our statistical graph knowledge representation. However, they apply without any loss of generality to all 1-variable line graphs contained in our larger knowledge base. Built with OWL technology, information retrieval across these statistical graphs goes beyond the query answering capabilities offered by i) original application tools, ii) XQuery over iGraphXML or iii) SPARQL in RDF triple stores. The expressive statistical graph ontology includes inverse, transitive and composed properties, negation, qualified cardinality restrictions and fully defined classes, all of which enable the discovery of implicit knowledge. The queries identified in the use cases require many of these OWL-DL exclusive features. The most effective way to query and retrieve statistical graph information requires the reasoning services of DL reasoners (e.g. Pellet [16], FACT++ [17], RacerPro [18]). Pellet and FaCT++ can be accessed via the OWL API while others can be queried using a proprietary API (i.e. RacerPro).

While there is no official recommendation for querying OWL ontologies, there exists several possibilities including DL class descriptions, RDF-based query languages (SPARQL, SeRQL, RQL) or the DL enhanced SPARQL-DL , or by using application dependent query languages such as RacerPro's nRQL. Using DL class descriptions for queries, members of that class will be retrieved, however, languages with variables that allow the retrieval of the instances of the model that satisfies the query are needed in some scenarios. For instance, we can formulate a query to an RDF/OWL knowledge base to retrieve the entire named graph and its parts. We might also extract any part thereof, provided that i) the reasoning services over the KB have the ability to infer the inverse of the heavily used **hasPart** and **isPartOf** properties, ii) only one relation is consistently asserted, or iii) both relations are explicitly added into the knowledge base. The XSLT used to transform iGraphXML to the SGO generally asserts one or the other of these relationships, and thus requires inferences over inverse properties for question answering.

An important aspect of our work involves the ability to semantically annotate the statistical graph at various levels of ontological commitment. This ability should provide the necessary framework by which informal tags can still be maintained within the semantic web framework, even in the absence of an ontology.

Importantly, tagging applications that seek out and instantiate existing ontologies can leverage these for more advanced logical reasoning.

8 Significance

Part of this work was inspired by the need of certain agencies to both exchange and integrate vast repositories of statistical graphs. Statistics Canada, for instance, publishes on average two communicative statistical graphs per day in their release of "The Daily". As previously discussed, searching over graphs that have been created with a given purpose in mind has more value added than just searching for the raw data in a database. Different statistical agencies in the world have expressed interest in exchanging semantically-annotated graphs to compare different datasets coming from data sources of statistical agencies around the world, to allow multilingual queries, etc. We are in the process of streamlining the publication of semantically annotated graphs, although currently this has to be done, for privacy issues, a day later than the original graph was published.

Our work creates new avenues for Semantic Science to grow and thrive. Semantic Science (s-Science) aims to capture and publish scientific data and scientific theories, thus making possible the selection and evaluation of competing theories, and their selection and use in knowledge discovery [19,20]. With Web 2.0 and Semantic Web technologies, new opportunities arise for knowledge management and discovery from scientific research in academia, government and industry. Our work provides a solid framework in which graphs and their components can be semantically annotated and referred to in support of or to dispute scientific hypothesis or claims. The large scale adoption of this knowledge representation for graphs will create new opportunities to mine scientific knowledge in pursuit of increased scientific understanding.

9 Future Work

A long term goal of the **gSem** project is to provide the tools to semantically annotate graphs while they are being designed. Towards this goal, we are augmenting iGraph plugins to allow users to tag their graphs with either informal tags or instances of existing ontologies. Use of more sophisticated schema matching algorithms will facilitate the addition of ontological knowledge. In addition, we are pursuing strategies that will allow the arbitrary annotation of any graph component such that the resulting instantiated ontology will provide author generated comments or tags. Semantic annotation of iGraph enriched graphs will also create new opportunities for the blind or visually impaired using the iGraph-Lite accessibility tool that represents and queries the different parts of a graph in order to render them accessible by a natural language interface.

A significant challenge exists in the automatic semantic annotation of existing graphs by means of natural language processing. The large body of Statistics Canada graphs are currently represented as unstructured images, and this will require integration of existing technologies to extract image-based labels with OCR

technologies, followed by natural language processing and novel algorithms to identify potential communicative meanings. iGraph performs a semi-automated annotation of the value axis and category axis values, but is limited to a fix set of keywords expected given the type of graphs analyzed (i.e. time series graphs).

10 Conclusion

In this paper we expand a knowledge representation of statistical graphs such that their semantic annotation is compatible with Web 2.0 tagging but can be extended to take advantage of the formalism afforded by the OWL-DL semantic web language. We demonstrate question answering and knowledge discovery across an example graph knowledge base that involve recent and highly expressive elements of the OWL language. Our work generates new opportunities for building semantically annotated statistical graphs that can be used beyond their communicative intent and may have a direct role in the newly evolving semantic science.

Acknowledgments. The authors would like to thank the anonymous reviewers who suggested valuable changes that improved the quality of this paper. This work was supported in part by an NSERC Discovery Grant for M. Dumontier and by a Statistics Canada research contract and a NSERC and Cognos Inc. Collaborative Research and Development grant (NSERC CRD 346381-06) to L. Ferres.

References

1. Cleveland, W.: The Elements of Graphing Data. Hobart Press, Summit (1994)
2. Lewandowsky, S., Spence, I.: The perception of statistical graphs. Sociological Methods and Research 18, 200–242 (1989)
3. Ecma International: Office Open XML File Formats (2006), http://www.ecma-international.org/publications/standards/Ecma-376.htm
4. Organization for the Advancement of Structured Information Standards (OASIS): Open Document Format for Office Applications (OpenDocument) v1.0 (2005), http://www.oasis-open.org/committees/download.php/12572/
5. Ferres, L., Dumontier, M., Villanueva-Rosales, N.: Semantic query answering with time-series graphs. In: Proceedings of the Eleventh International IEEE EDOC Conference Workshop, Annapolis, MD, USA, pp. 117–124. IEEE Computer Society Press, Los Alamitos (2007)
6. Ferres, L., Verkhogliad, P., Lindgaard, G., Boucher, L., Chrétien, A., Lachance, M.: Improving accessibility to statistical graphs: the igraph-lite system. In: Assets 2007: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility, pp. 67–74. ACM Press, New York (2007)
7. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation, and Applications. In: Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.) Description Logic Handbook. Cambridge University Press, Cambridge (2003)
8. Larkin, J., Simon, H.: Why a diagram is (sometimes) worth ten thousand words. Cognitive Science 11(1), 65–100 (1987)

9. Artale, A., Franconi, E.: A survey of temporal extensions of description logics. Annals of Mathematics and Artificial Intelligence 30(1-4), 171–210 (2000)
10. Artale, A., Franconi, E.: Introducing temporal description logics. In: TIME 1999: Proceedings of the Sixth International Workshop on Temporal Representation and Reasoning, Washington, DC, USA, p. 2. IEEE Computer Society, Los Alamitos (1999)
11. Ester, M., Frommelt, A., Kriegel, H.P., Sander, J.: Algorithms for characterization and trend detection in spatial databases. In: Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD), pp. 44–50 (1998)
12. Haarslev, V.: Formal semantics of visual languages using spatial reasoning. In: VL 1995: Proceedings of the 11th International IEEE Symposium on Visual Languages, Washington, DC, USA, p. 156. IEEE Computer Society, Los Alamitos (1995)
13. Pinker, S.: A theory of graph comprehension. In: Freedle, R. (ed.) Artificial intelligence and the future of testing, pp. 73–126. L. Erlbaum, Hillsdale (1990)
14. Parsia, B., Patel-Schneider, P.F.: Owl 2 web ontology language: Primer. World Wide Web Consortium (W3C) Working Draft (2008)
15. Horridge, M., Drummond, N., Goodwin, J., Rector, A., Stevens, R., Wang, H.H.: The manchester owl syntax. In: Proceedings of the OWL: Experiences and Directions Workshop Series (2006)
16. Sirin, E., Parsia, B.: Pellet: An owl dl reasoner. In: Haarslev, V., Möller, R. (eds.) Proceedings of the 2004 International Workshop on Description Logics (DL 2004), Whistler, British Columbia, Canada. CEUR Workshop Proceedings, vol. 104, CEUR-WS.org (2004)
17. Tsarkov, D., Horrocks, I.: Fact++ description logic reasoner: System description. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 292–297. Springer, Heidelberg (2006)
18. Haarslev, V., Möller, R., Wessel, M.: Querying the semantic web with racer + nrql. In: Proceedings of the KI-2004 International Workshop on Applications of Description Logics (ADL 2004), Ulm, Germany, September 24 (2004)
19. Poole, D.: Semantic science: machine understandable scientific theories and data (2007), www.cs.ubc.ca/spider/poole/Semantic-Science/SemanticScience.pdf
20. Hendler, J.: Science and the semantic web. Science 299(5606), 520–521 (2003)

Towards a Methodology to Conceptualize the Geographic Domain

Miguel Torres, Rolando Quintero, Serguei Levachkine, Giovanni Guzmán, and Marco Moreno

Intellingent Processing of Geospatial Data Lab-Centre for Computing Research-National Polytechnic Institute, Mexico City, Mexico
{mtorres,quintero,sergei,marcomoreno,jguzmanl}@cic.ipn.mx
http://geo.cic.ipn.mx http://www.geosco.org

Abstract. To date, there are different ontologies for many domains and applications. Users can access them in order to share information, reuse knowledge and integrate data sources for several purposes such as semantic web, data warehouse, e-learning, e-commerce, knowledge representation and so on. Ontology engineering is rapidly becoming a mature discipline, having produced tools and methodologies for building and managing ontologies. However, even with a clearly defined engineering methodology, building a large ontology remains a challenging, time-consuming and error-prone task, since it forces ontology builders to conceptualize their expert knowledge explicitly and to re-organize it in typical ontological categories such as concepts, properties and axioms. An approach to conceptualizing the geographic domain is described. It is oriented to formalize the geographic domain conceptualization according to specifications from the INEGI[1]. The main goal is to provide semantic and ontological descriptions, which represent the properties and relations that describe the behavior of geographic objects by means of concepts. GEONTO-MET is focused on developing geographic application ontologies for the sharing and integrating of geospatial information.

1 Introduction

Ontologies are widely used in Knowledge Engineering, Artificial Intelligence and Computer Science, in applications related to knowledge management, natural language processing, e-commerce, intelligent information integration, information retrieval, database design, bio-informatics, education, and in new emerging fields like the Semantic Web. An ontology is an explicit formal specification of terms, which represents the intended meaning of concepts, in the domain and relations among them. The primary goal of an ontology is sharing and reusing knowledge among people or machines [1]. It is a crucial issue for the success of many knowledge-based applications and intelligent systems [2]. Because it is difficult to find concrete ontologies within real-world knowledge domain, the development of ontologies requires huge efforts. At this time, there are different and public ontologies for many domains

[1] Instituto Nacional de Estadística, Geografía e Informática.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 111–122, 2008.

and applications. Users can access them in order to share information, reuse knowledge and integrate data sources for several purposes. From this point of view, ontology engineering is rapidly becoming a mature discipline which has produced various tools and methodologies for building and managing ontologies. However, even with a clearly defined engineering methodology, building a large ontology remains a challenging, time-consuming and error-prone task, since it forces ontology builders to conceptualize their expert knowledge *explicitly* and to *re-organize* it in typical ontological categories such as concepts, properties and axioms [1]. Knowledge acquisition and reuse are often seen as ways to make this tedious process more efficient: even though both methods cannot *currently* be used to *automatically* generate a domain ontology satisfying a specific set of requirements, they can be used to guide or accelerate the modeling process [3]. Ontology has gained attention among researchers in geospatial information science in recent years. Ontology can play an important role in establishing robust theoretical foundations for geographic information science. In [4] is sketched out three broad sets of foundational: (i) conceptual issues concerning what would be required to establish an exhaustive ontology of the geospatial domain, (ii) representational and logical issues relating to the choice of appropriate methods for formalizing ontologies, and (iii) issues of implementation regarding the ways in which ontology ought to influence the design of information systems.

In this paper, we have focused on the first and second issues in order to develop the GEONTO-MET approach to conceptualizing the geographic domain. This methodology allows for *integrating* and *sharing* geospatial information as well as for avoiding the *ambiguities* that are sometimes present in geographical data. Many questions can be asked such as: why is it important to conceptualize the geographic domain? What are the main goals for building approaches oriented to generate geo-ontologies? What are the main benefits in conceptualizing this domain? The answers are directly focused on describing the limitations that *numeric focus* offers in this context (geographic domain). Several implicit characteristics of geospatial information are related to the scale, projection, and reference systems. These have produced problems associated with the semantic heterogeneity, interoperability, data sharing, information integration and imprecise data processing. For instance, if we ask where the Eiffel Tower is, the answer, of a geographic information system is, *"in coordinates 48º 51' 30" N; 2º 17' 40" W"*. However, if we ask the same question to a person, his answers could be "in Paris", "near Avenue des Champs-Elysées" or "on the Rive Gauche". The main difference between these answers (*GIS* and *people*) is that the thought of a human being is *conceptual* and the *reasoning* is directly related to the *cognitive system*. Then, numerical methods are not common in our thoughts to find answers about this kind of questions.

Therefore, this work is oriented towards processing geospatial data in a *semantic focus*. This research assumes that when we, as human beings *think spatially*, we relate locations to our cognitive system for reasoning and describing the geographic environment. Features like coordinates, projections, spatial databases or maps are not considered in our mind maps. We think about things that give us a reference associated with different *labels*, in which a location that is well known provides a "*description*" or a reference. It is widely recognized that *semantics* of geospatial information is crucial to the development of interoperable geospatial data and software. It is also widely recognized that GIS software and technology should be able to interoperate with other applications and databases. Such interoperability requires a common

shared ontology for the phenomena under consideration. This means also, that approaches to the conceptualization of the geographic domain provide common and shared ontological descriptions designed to establish geospatial specifications or terminology standards to partially solve this interoperability.

In summing up, the *integration, sharing* and *interoperability* of geospatial information is the main goal of the GEONTO-MET methodology. Semantic descriptions are built to achieve these purposes. In addition, *Kaab-ontology* has been designed using this approach.

The remainder of the paper is structured as follows: Section 2 briefly reviews earlier methodologies for building ontologies and works oriented to semantics in the geographic domain, Section 3 describes the GEONTO-MET approach, which is proposed to conceptualize the geographic domain. Conclusions and suggestions for future work are outlined in Section 4.

2 Related Works

Basically, a set of methods and methodologies for developing ontologies has been reported. In [5] some general steps and interesting points about Cyc development were published. Cyc is a huge knowledge base (KB) with common sense knowledge. It was built upon a core of over 1,000,000 hand entered assertions designed to capture a large portion of what people normally consider consensus knowledge about the world. During the development of Cyc the following processes were carried out: 1) manual coding of articles and pieces of knowledge, 2) knowledge coding aided by tools using the knowledge already stored in the Cyc KB, and 3) knowledge coding mainly performed by tools using knowledge already stored in the Cyc KB. Cyc is considered an ontology, because it can be used as a substrate for building different intelligent systems that can communicate and interact with others.

In [6] is described the first method for building ontologies, which was extended in [7]. Authors proposed some guidelines based on their experience in developing the Enterprise Ontology. This ontology was developed as a part of the Enterprise Project. To build an ontology, according to Uschold and King's approach, the following processes must be performed 1) identify the purpose of the ontology, 2) build it, 3) evaluate it, and 4) document it. During the building process, authors proposed capturing knowledge, coding it and integrating other ontologies inside the current ontology.

Based on TOVE (TOronto Virtual Enterprise) project, in [8] was published a formal approach to building and evaluating ontologies. This methodology has been used to build the TOVE ontologies. It is inspired by the development of knowledge based systems using first order logic. They proposed identifying intuitively the main scenarios, that is, possible applications in which the ontology will be used. Then, a set of natural language questions, called competency questions are used to determine the scope of the ontology. These questions and their answers are both used to extract the main concepts and their properties, relations and formal axioms of the ontology. This is a very formal methodology that takes advantage of the robustness of classic logic and can be used as a guide to transform informal scenarios in computable models.

In [9] the KACTUS approach to investigating the feasibility of knowledge reuse in complex technical systems and the role of ontologies to support it is proposed. It is

conditioned by application development. Thus, every time an application is built, the ontology that represents the knowledge required for the application is refined. The ontology can be developed by reusing others and can be integrated in ontologies of later applications. Therefore, the following processes occur every time an application is developed: 1) specification of the application, 2) preliminary design based on relevant top-level ontological categories, and 3) ontology refinement and structuring.

The METHONTOLOGY approach [10] proposes the enabling of the construction of ontologies at the knowledge level. It has its basis in the main activities identified by the software development process and in knowledge engineering methodologies. This methodology includes: the identification of the ontology development process, a life cycle based on evolving prototypes, and methods to carry out each activity in the management, development-oriented and support tasks. The main processes involved in this approach are the following: 1) build the glossary of terms, 2) build concept taxonomies, 3) build ad hoc binary relation diagram, 4) build the concept dictionary, 5) define ad hoc binary relations, 6) define instance attributes, 7) define class attributes, 8) define constants, 9) define formal axioms, and 10) define rules. The SENSUS based-method is proposed in [11] for building the skeleton of the domain ontology, starting from a huge ontology. The SENSUS links domain-specific terms to the huge ontology and prunes, in the huge ontology, those terms that are irrelevant for the new ontology. The result of this process is the skeleton of the new ontology, which is automatically generated using the following tasks: 1) identify seed terms, 2) manually link the seed terms to SENSUS, 3) add paths to the root, 4) add new domain terms, and 5) add complete subtrees.

The aim of On-To-Knowledge [12] methodology is to apply ontologies to electronically available information in order to improve the quality of knowledge management in large and extended organizations. This approach proposes to build the ontology taking into account how the ontology will be used in further applications. Consequently, ontologies developed with this methodology are highly dependent on the application. As a characteristic, it proposes ontology learning for reducing the efforts made to develop the ontology. The processes of this approach are the following: 1) feasibility study, 2) kickoff, 3) refinement, 4) evaluation, and 5) maintenance.

In [13], Guarino has proposed the Ontoclean, as a method to analyze and clean the taxonomy of an existing ontology by means of a set of principles based on philosophy. It is oriented to remove wrong *Subclass-Of* relationships in taxonomies, according to some philosophical notions such as *rigidity*, *identity* and *unity*. These notions are applicable to properties, but they can be extended to concepts.

Some works related to ontologies and semantics in geospatial information science to be mentioned are as follows:

Smith *et al.* [14] reported the results of a series of experiments designed to establish how non-expert subjects conceptualize geospatial phenomena. Subjects were asked to give examples of geographical categories in response to a series of differently phrased elicitations. The results yielded an ontology of geographical categories – a catalogue of the prime geospatial concepts and categories shared in common by human subjects, independently of their exposure to scientific geography. Mark, *et al.,* [15] designed an ontology of geographic kinds to yield a better understanding of the structure of the geographic world, and to support the development of GIS that is conceptually sound. This work first demonstrated that geographical objects and kinds are

not only larger versions of the everyday objects and kinds previously studied in cognitive science.

Methodologies and approaches described in this section are used with several purposes: to create a new ontology from scratch, to enrich an existing one with new terms, and to acquire knowledge for some tasks. Ontologies aim to capture *consensual* knowledge of a given domain in generic and formal ways to be reused and shared across applications and by groups of people.

3 GEONTO-MET Methodology

GEONTO-MET methodology [16-18] is oriented to formalize geographic domain conceptualization according to the specifications from the INEGI. The main goal is to provide semantic descriptions, which represent the properties and relations being described so that the behavior and features of geographic objects are taken into account directly from the geographic domain ontology.

GEONTO-MET is composed of four principal stages: *Analysis* provides an abstract model of the geographic objects involved in this domain. *Synthesis* makes the conceptualization of the geographic domain. A set of application ontologies (in tourist and topographic contexts) and domain ontology called *Kaab-Ontology* is generated by the *Processing* stage. Finally, *Description* produces an alternative representation of geographic objects as well as the integration of them in a semantic description template. This approach is based on a set of axiomatic relationships allowing for direct translation of the relations between concepts to the conceptualization. In this way, the *semantic resolution* is improved, that is, the definition of such relationships can be iteratively refined. To achieve this, we use a couple of sets ($A_1 = \{is, has, does\}$ and A_2 specific prepositions related to geospatial context). These sets are necessary and sufficient to define the rest of relationships, involved in the conceptualization of geographic domain.

The essence of this methodology is to reduce the *axiomatic relationships* within conceptualization. One could think that this reduction is a limitation for the richness of expression that conceptualization can implicitly contain. Nevertheless, the universe of possible relations is not *a priori* defined, due to the *"relation"* in a classic sense is not predefined. In fact, the reduction of axiomatic relations has two main advantages: the first, it is possible to define as many *"typical relations"* as needed, because this type of relation is treated as a concept. In other words, *"typical relations"* are part of conceptualization, they are not considered as axioms, and these are defined as *concepts*. The second advantage is that relations have a semantic association to themselves, not only from an axiomatic definition, but also from the conceptualization itself (the context of each relation).

To illustrate the idea, let us consider one axiomatic relation widely used: the relationship *"part_of"*. Such relationship means that one concept is a constituent element of another concept. By using GEONTO-MET, it is possible to create this relationship as a concept *(concept-R)*, by defining the concept *"part"* (in the way that the concepts are defined) and using the axiomatic relationships (*is*, *does*, *has*) to create the *concept-R* equivalent to the relationship *"part_of"*. For instance, let's consider the following sentence: "heart *part_of* body"; in this, two concepts (*heart* and *body*) are involved as

well as one axiomatic relationship (*part_of*). By using the methodology, the same relationship could be expressed "heart *is* part of body", in which three concepts (*heart*, *part* and *body*) and two axiomatic relationships: a fundamental one (*is*) and an auxiliary one (*of*) are described. The advantage is that the semantic of the relationship is the same and it is not necessary to previously define the relation *"part_of"*. Then, we only need to define *"part"* as a concept (having the semantic richness of concepts) and use it to define the new relation. On the other hand, GEONTO-MET is composed of a set of elements that are used to make the *geographic domain conceptualization*. They are described in the next subsections.

3.1 Axioms in GEONTO-MET

This methodology minimizes the number of axioms by means of the reduction of the axiomatic relationships. In order to do this, we have defined a small set of axiomatic binary relationships, divided into two subsets. The first, subset A_1 contains three relations that will be called *fundamental relations*, and it is denoted by equation 1.

$$A_1 = \{is, has, does\} \tag{1}$$

The *"is"* relationship means an *existence* or identity, such it is used to characterize the concepts in the conceptualization. It implies the inheritance of properties and abilities. Also, it allows for making a hierarchy of the same concepts. Some names for this relationship could be *"son-of"* or, *"is-a"*. As binary relation, *"is"* has the following properties. Let C be a set of concepts:

- **It is anti-symmetric:** $\forall a,b \in C, a(is)b \wedge b(is)a \Rightarrow a = b$, if concept A *"is"* B and B *"is"* A, necessarily A and B are the same concept. For instance, "lake" *"is"* "waterbody" but "waterbody" *("is")* not lake, because there are other geographic concepts that are also "waterbody".
- **It is reflexive:** $\forall a \in C, a(is)a$, each concept A *"is"* itself. For instance, "Chapultepec Lake" *"is"* "Chapultepec Lake", because a geographic object has its own identity.
- **It is transitive:** $\forall a,b,c \in C, a(is)b \wedge b(is)c \Rightarrow a(is)c$, if A *"is"* B and B *"is"* C then A *"is"* C. For instance, if "road" *"is"* "og_artificial" and "og_artificial" *"is"* "geographic object", then "road" *"is"* "geographic object".

The *"has"* relationship describes *aggregation* or *association*, such it is used to define the properties that build up a concept. The properties of this binary relation are as follows:

- **It is non-symmetric:** $\exists a,b \in C, a(has)b \wedge b(has)a \Rightarrow a = b$, there are concepts A and B so that if A *"has"* B and B *"has"* A, then A and B are the same concept. For instance, "state" "has" "counties" but "counties" do not have (*"has"*) "states".
- **It is irreflexive:** $\exists a \in C \ni a \neg (has)a$, there is some concept A such that it does not have itself as property. For instance "urban area" does not have (*"has"*) "urban area".

- **It is transitive:** $\forall a,b,c \in C, a(has)b \wedge b(has)c \Rightarrow a(has)c$, if A *"has"* B and B *"has"* C, then A *"has"* C. For instance, if "country" *"has"* "state" and "state" *"has"* "county", then "country" *"has"* "county".

The *"does"* relationship is used to describe an action, such it defines the abilities or operations associated with a concept. Its properties are as follows:

- **It is symmetric:** $\exists a,b \in C, a(does)b \wedge b(does)a \Rightarrow a = b$, there are concepts A and B such that if A *"does"* B and B *"does"* A, then A and B are the same concept.
- **It is irreflexive:** $\exists a \in C \ni a \neg(does)a$, there is some concept A that does not have itself as an ability.
- **It is non-transtive:** $\exists a,b,c \in C \ni a(does)b \wedge b(does)c \Rightarrow \neg a(does)c$, if A *"does"* B and B *"does"* C it does not necessarily imply that A *"does"* C.

The second subset of axiomatic relationships is denoted by A_2 and it is composed of prepositions. See the equation 2.

$$A_2 = \begin{Bmatrix} to, before, under, with, against, of, from, in, between, towards, until, \\ for, by, since, on, after, behind, beside, near, through \end{Bmatrix} \quad (2)$$

These relationships are defined as *asymmetric*, *irreflexive* and *non-transitive*, although linguistically some of them do not accomplish such properties[2]. For implementation convenience, we have considered that relationships in A_2 are defined as follows. Let $r \in A_2$ be a relationship, then, i) $\forall a,b \in C, arb \wedge bra \Rightarrow a = b$, ii) $\exists a \in C \ni a \neg ra$ and iii) $\exists a,b,c \in C \ni arb \wedge brc \Rightarrow a \neg rc$.

3.2 Relations in GEONTO-MET

Relations in GEONTO-MET can be classified in two types:

- **Simple:** This type has the form: $a\rho b \in R_S$, where $a,b \in C$ and $\rho \in A_1$.
- **Complex:** It has the form: $a\rho b\pi c \in R_C$, where $a,b,c \in C$ y $\rho \in A_1$ and $\pi \in A_2$.

As we have previously mentioned, *"is"* is a hierarchy relationship that provides the mechanism of *"inheritance"*. By using this relationship, we will be able to create a hierarchy of concepts (existence). The *"has"* relationship provides the capability of *"aggregation"* or *"composition"* of concepts. The *"does"* relationship describes the actions of a concept.

Relationships in A_2 allow for describing and providing *causality* or *intention* to the relationships in A_1. The complex form of relations is used to have a minimum number of axiomatic relationships and give us the semantic to non-axiomatic relationships. R_V describes the set of valid relationships. R_V is the set of all relationships such that comply with restrictions. On the other hand, R_R represents the set of relationships existing in real or concrete conceptualization.

[2] For instance, the relationship *"beside"* is a symmetric relation.

3.3 Concepts in GEONTO-MET

We have defined a concept as a collection of abilities and properties that share a single existence. There are four types of concepts:

- **Relational concepts** (verbs). They are defined as elements denoting an action or operation over other concepts. C_R denotes the set of all concepts-R.
- **Standard concepts** (noun). They are defined as elements belonging to a class. All their abilities and properties are abstract. C_E (concepts-E) denotes the set of this type of concept.
- **Class concepts**. They are concepts-E that allow for making partitions of C_R and C_E. They are denoted by C_L (concepts-L).
- **Instance concepts.** They are concepts whose abilities and properties are concrete.

Also, we can stand that $C = C_R \bigcup C_E \bigcup C_L$, such that $C_E \cap C_R = \varnothing$, $C_E \cap C_L = \varnothing$ and $C_L \cap C_R = \varnothing$, i.e. C_L, C_R and C_E are disjoined sets. An *abstract* concept is a concept-E, which does not have instances.

3.4 Properties in GEONTO-MET

Properties are concepts aggregated to one another by means of a relationship of belonging. These define the characteristics of the second concept. A property can be defined as follows: Let $a,b \in C$ be concepts, we say that b is a property of a if $a(has)b \in R_R$. b is a concrete property of a if b is an instance. It is called *abstract property* to the one that is not concrete. For instance, if "mountain" *"has"* "altitude", and "altitude" could be "low", "mid" and "high"; then "altitude" is a property of "mountain" and "low" is a concrete property of "mountain".

Π is the set of properties, Π_C is the set of concrete properties and Π_A is the set of abstract properties. So, in this way we can stand that $\Pi = \Pi_A \bigcup \Pi_C$ and $\Pi_A \bigcap \Pi_C = \varnothing$.

3.5 Abilities in GEONTO-MET

Abilities are concepts that define actions or operations associated with other concepts. As well, they describe how a concept interacts with other concepts. An ability is defined as follows: Let $a,b \in C$ be concepts, so we can say that b is an ability of a if $a(does)b \in R_R$. b is a concrete ability of a if b is an instance. The non-concrete abilities are called *abstract abilities*. H is the set of abilities, H_C is the set of concrete abilities and H_A is the set of abstract abilities. Also, $\mathrm{H} = \mathrm{H}_A \bigcup \mathrm{H}_C$ and $\mathrm{H}_A \bigcap \mathrm{H}_C = \varnothing$.

3.6 Instances in GEONTO-MET

Instances are a collection of concrete abilities and properties that have a unique existence. In other words, an *instance* is a concept contained within a hierarchy formed by the *"is"* relationship, whose properties and abilities are concrete. Thus, an instance is a concept with no sons within the hierarchy, - it is not a class and all its abilities and properties are instances. I represents the set of all instances in a conceptualization[3].

3.7 Classes in GEONTO-MET

Classes are concepts that have *children concepts* by means of the existence relationship (*"is"*), their sons cannot be an *instance*, and they form a *complete partition*.

In this case, the *"is"* relationship is equivalent to the relationship *"subclass_of"*. C_L denotes the set of classes.

3.8 Constraints in GEONTO-MET

A constraint is described as a *sentence* to avoid inconsistencies in the definition and description of the elements involved in geographic domain conceptualization. A set of constraints defined in GEONTO-MET are sketched out as follows.

i) Constraints for relations in the simple form

Let $a,b \in C$ be concepts, then $a(is)b \in R_V$ if,

- $a(is)b \in R_S \Rightarrow b(is)a \notin R_V$, for avoiding redundant inheritances, because *"is"* relationship is reflexive. This restriction eliminates $\left[\begin{matrix} n \\ 2 \end{matrix}\right] \Big/ 2$ possible relationships, where $n = card(C)$.
- $a(is)b, b(is)c \in R_S \Rightarrow a(is)c \notin R_V$, for avoiding redundant inheritances, because *"is"* relationship is transitive.
- $a(is)b \in R_S \Rightarrow b \notin I$, instances do not have sons. This restriction eliminates $n_i n$ possible relationships, where $n_i = card(I)$.
- $a(is)b \in R_S, b \in C_L \Rightarrow a \notin I$, instances cannot inherit from classes. This restriction eliminates $n_i n_l$ possible relationships, where $n_l = card(C_L)$.

Similarly, the *"has"* relationship contains restrictions in its simple form; Let $a,b \in C$ be concepts, then $a(has)b \in R_V$ if,

- $a(has)b \in R_S \Rightarrow a \notin C_L$, classes do not have properties. This restriction eliminates $n_i n$ possible relationships.
- $a(has)b \in R_S, a \in I \Rightarrow b \notin C_L$, classes cannot be properties of instances. This restriction eliminates $n_i n_l$ possible relationships.
- $a(has)b \in R_S \Rightarrow b \notin I$, instances cannot be properties of non-concrete concepts. This restriction eliminates $n_i n$ possible relationships.

[3] In a pragmatic form I is a database.

The *"does"* relationship has its restrictions when it is used in its simple form; Let $a,b \in C$ be concepts, then $a(does)b \in R_V$ if,

- $a(does)b \in R_S \Rightarrow a \notin C_L$, classes do not have abilities. This restriction eliminates $n_l n$ possible relationships.
- $a(does)b \in R_S, a \in I \Rightarrow b \notin C_L$, classes cannot be abilities of instances. This restriction eliminates $n_i n_l$ possible relationships.
- $a(does)b \in R_S \Rightarrow b \notin I$, concepts cannot have instances as abilities. This eliminates $n_i n$ possible relationships.

ii) Constraints for relations in the complex form

Let $a,b,c \in C$ be concepts, and $\pi \in P$, then:

- $a(is)b\pi c \notin R_V$ if,
  - $a(is)b\pi c \in R_C, a \in C_E \Rightarrow b \notin I$, instances cannot have sons.
  - $a(is)b\pi c \in R_C, a \in I \Rightarrow b \in C_E$, instances can only be sons of concepts-E.
  - $a(is)b\pi c \in R_C \Rightarrow b \notin I$, *"is"* relation can be used for describing ternary relations, if the concept-R is not an instance.
- $a(has)b\pi c \in R_V$ if,
  - $a(has)b\pi c \in R_C \Rightarrow b \in C_E$, *"has"* relationship can be used for describing ternary relationships, if the middle concept is a concept-E.
  - $a(has)b\pi c \in R_C \Rightarrow a \notin C_L$, classes do not have properties.
  - $a(has)b\pi c \in R_C, \pi \in P \Rightarrow c \notin I$, concepts cannot have instances as properties.
  - $a(has)b\pi c \in R_C, a \in I, \pi \in P \Rightarrow c \notin C_L$, instances do not have classes as properties.
- $a(does)b\pi c \in R_V$ is,
  - $a(does)b\pi c \in R_C, \pi \in P \Rightarrow b \in C_R$, *"does"* relationship can be used for describing ternary relationships if the middle concept is a concept-R.
  - $a(does)b\pi c \in R_C, \pi \in P \Rightarrow a \notin C_L$, classes do not have abilities.
  - $a(does)b\pi c \in R_C, \pi \in P \Rightarrow c \notin I$, concepts cannot have instances as abilities.

4 Conclusion and Future Work

In this paper, a description of GEONTO-MET methodology has been outlined. However, we have only focused on the task of *conceptualization* (Synthesis stage). The advantage of GEONTO-MET is being able to a general framework based on abstract classes to classify and store geographic concepts, defined in a specific context, as well

as the explicit relationships. This fact is used to extract the *semantics* that offers the mapping between concepts through their relationships. The *constraints* defined in *Kaab* play an important role to validate the definition of a concept, according to the relationship that corresponds to it. The minimization of axiomatic relationships is focused on *enriching* the conceptualization, directly translating the relationships of the concepts to the ontology. This mechanism offers a higher *semantic granularity* to the conceptualization and it is more flexible for generating ontological descriptions.

The methodology has been designed with a general vision of knowledge representation. It is necessary to do more tests in order to determine its use in other domains. At this time, the geographic domain is totally covered by this approach. Additionally, GEONTO-MET can be compared to the *object-oriented* model; therefore any kind of geographic phenomenon can be ontologically modeled with this approach. We have defined geographic domain ontology as follows: "it is a logical and defined theory in a formal language and represented by an explicit vocabulary, which is based on two sets of axiomatic relationships that are used to reason and understand the entities of the geographic world that can be conceptualized".

On the other hand, this approach allows for *integrating* and *sharing* geospatial information. It provides feasible solutions towards these and other related issues. Additionally, by using geographic domain conceptualization it is possible to avoid the *ambiguity* of terms, which represent the *semantics* of the geographic domain in the ontology.

Future works are focused on establishing the guidelines to make *spatial reasoning*. That is, if the behavior of geographic objects involved in an ontology partition is known, then one can understand the *semantics* that they represent and explore the implicit *inference* of the conceptualization. On the other hand, these approaches for processing geospatial data with a *semantic focus*, establish the basis and requirements to produce in short time *iGIS* (intelligent geographic information systems).

Acknowledgments

The authors of this paper wish to thank the CIC, SIP, IPN and CONACYT for their support.

References

1. Gruber, T.: From A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2), 199–220 (1993)
2. Guarino, N.: Formal Ontology and Information Systems. In: Proceedings of the International Conference on Formal Ontology in Information Systems, FOIS 1998, Trento, Italy, pp. 3–15 (1998)
3. Gómez, A., Fernández, M., Corcho, O.: Ontological Engineering, 2nd edn. Springer, Heidelberg (2004)
4. Mark, D.M., Smith, B., Egenhofer, M., Hirtle, S.: Emerging Research Theme: Ontological Foundations for Geographic Information Science, University Consortium for Geographic Information Science, Technical Report (2001)

5. Lenat, D.B., Guha, R.V.: Building Large Knowledge-based Systems: Representation and Inference in the Cyc. Project. Addison-Wesley, Boston (1990)
6. Uschold, M., King, M.: Towards a Methodology for Building Ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing IJCAI 1995, Montreal, Canada, pp. 6.1–6.10 (1995)
7. Uschold, M., Grüninger, M.: Ontologies: Principles, Methods and Applications. Knowledge Engineering Review 11(2), 93–155 (1996)
8. Grüninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing IJCAI 1995, Montreal, Canada, pp. 7.3–7.13 (1995)
9. Bernaras, A., Laresgoiti, I., Corera, J.: Building and reusing ontologies for electrical network applications. In: Proceedings of European Conference on Artificial Intelligence, Budapest, Hungary, pp. 298–302. John Wiley & Sons, Chichester (1996)
10. Fernández, M., Gómez, A., Juristo, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: Symposium on Ontological Engineering of AAAI, pp. 33–40. Standford University, California (1997)
11. Swartout, B., Ramesh, P., Knight, K., Russ, T.: Toward Distributed Use of Large-Scale Ontologies. In: Symposium on Ontological Engineering of AAAI, pp. 138–148. Standford University, California (1997)
12. Staab, S., Schnurr, H.P., Studer, R., Sure, Y.: Knowledge Processes and Ontologies. IEEE Intelligent Systems 16(1), 26–34 (2001)
13. Guarino, N., Welty, C.: A Formal Ontology of Properties. In: Dieng, R., Corby, O. (eds.) EKAW 2000. LNCS (LNAI), vol. 1937, pp. 97–112. Springer, Heidelberg (1995)
14. Smith, B., Mark, D.: Ontology and Geographic Kinds. In: Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, Canada, pp. 308–320 (1998)
15. Mark, D., Smith, B., Tversky, B.: Ontology and Geographic Objects: An Empirical Study of Cognitive Categorization. In: Proceedings of the Spatial Information Theory: A Theoretical Basis for GIS, Stade, Germany, pp. 283–298 (1999)
16. Torres, M., Levachkine, S.: Obtaining semantic descriptions based on conceptual schemas embedded into a geographic context. Lecture Notes in Geoinformation and Cartography, vol. 4977, pp. 209–222. Springer, Berlin (2007)
17. Torres, M., Moreno, M., Quintero, R., Fonseca, F.: Ontology-driven description of spatial data for their semantic processing. In: Rodríguez, M.A., Cruz, I., Levashkin, S., Egenhofer, M.J. (eds.) GeoS 2005. LNCS, vol. 3799, pp. 242–249. Springer, Heidelberg (2005)
18. Torres, M.: Representación ontológica basada en descriptores semánticos aplicada a objetos geográficos. Ph. D. Thesis in Spanish (2007)

Effect of Preprocessing on Extractive Summarization with Maximal Frequent Sequences

Yulia Ledeneva

National Polytechnic Institute
Center for Computing Research
Av. Juan de Dios Bátiz s/n, D.F., 07738, Mexico
yledeneva@yahoo.com

Abstract. The task of extractive summarization consists in producing a text summary by extracting a subset of text segments, such as sentences, and concatenating them to form a summary of the original text. The selection of sentences is based on terms they contain, which can be single words or multiword expressions. In a previous work, we have suggested so-called Maximal Frequent Sequences as such terms. In this paper, we investigate the effect of preprocessing on the process of selecting such sequences. Our results suggest that the accuracy of the method is, contrary to expectations, not seriously affected by preprocessing—which is both bad and good news, as we show.

1 Introduction

The importance of preprocessing procedure is evident because it is used in almost every developed system related with text processing.

Preprocessing includes one or more of the following steps. The first step is lexical analysis of the text with the objective of treating digits, hyphens, punctuation marks, and the case of letters. In the second step, semantically empty words (stopwords) can be eliminated. Finally, stemming can be applied which comprehends grouping similar words together by means of small changes: for example, eliminating prefixes and suffixes of words.

The task of extractive text summarization consists in communicating the most important information of a document. Usually, linguistic, statistic or heuristic methods are used to extract the most important parts of a text and use them as new terms. These terms must be more precise and shorter than original text. Moreover, the terms tend to reflect the content in the best possible manner. Such terms can be of different granularity: words or multiword terms such as ngrams or whole phrases. In this sense, the terms which are composed of several words can be called multiword descriptions.

Taking in consideration that the importance of a part of text is determined based on terms it contains, the extraction of such terms (we denote them as multiword descriptions) was proposed in [1]. As such multiword descriptions, the Maximal Frequent Sequences (MFSs) was extracted in [1, 2]. In those works, it was shown that with MFSs it is possible to represent important information mentioned in the text facilitating automatic generation of text summaries. In this work, we apply preprocessing with the objective to detect whether it influences on obtaining better extractive text summaries.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 123–132, 2008.

In this paper, we consider single text summarization, more specifically the method of automatic detection of multiword descriptions. The preprocessing procedure involves various steps in order to prepare the text for a particular task. Here, this particular task is the extraction of multiword descriptions. More specifically, we compare text summaries generated using the method based on extraction of multiword descriptions before and after preprocessing.

The paper is organized as follows. Section 2 summarizes the related work on the influence of preprocessing. In Section 3, typical text preprocessing procedure is described in details. The general scheme of the proposed method is described in Section 4. The experimental results are presented in Section 5. Section 6 concludes the paper.

2 Related Work

Typically, each system which deals with texts involves a stage of preprocessing. Unfortunately, there are few works dedicated to research how preprocessing influences on a specific task. The work [3] shows how preprocessing improve the task of text categorization. There are several works [4–7] done about preprocessing in information retrieval (IR). The importance of stemming for performance of IR systems is studied in [4]. The authors of [5-7] illustrate that for effective IR a highly inflected language requires good stemming.

While arguments supporting stemming seem sensible, there is a controversy in the literature about the benefits of stemming for text processing tasks. In fact, different studies lead to rather contrasting conclusions. In [8] eight studies on the potential benefits of stemming are compared. However, the use of stemming are favored in [8], the results of eight experimental studies reported in the same work do not permit to reach a decisive conclusion. As a result of this controversy, many text processing procedures do not adopt any stemming algorithm whatsoever.

In our previous work [1, 2], we have tested different combination of different text summarization steps for single text summarization. These steps are term selection, term weighting, sentence weighting, and sentence selection. For term selection, one should decide what the terms should be used: for example, words, ngrams [9], basic elements [10], or other semantic units [11, 12]. A big advantage of the use of these terms is their automatic extraction. In term weighting, the usefulness of a terms are estimated based on machine learning [9] or methods using lexical and semantic information [13]. For sentence weighting, commonly the weight of each sentence is calculated using the weight of all its terms. And finally, in the step of sentence selection, the best sentences are selected based on the numerical measure calculated in previous steps. In [14] most sophisticated methods are used for sentence selection.

It was concluded that MFSs and words derived from MFSs are good terms for the generation of single text summaries in [1]. However, preprocessing was not applied that has been shown beneficial for other summarization methods [14, 15]. In this work, our hypothesis is that use of preprocessing for extracting MFSs gives better results than extracting MFSs without preprocessing.

3 Text Preprocessing Procedure

A typical preprocessing procedure consists in several steps [16], each of these steps can or can't be done depending on whether the user it requires. Also, these steps are called text operations or transformations. Namely, we consider the following steps of preprocessing procedure:

– *Lexical analysis*. The main objective of lexical analysis process is the identification of the words in a text. Usually, the following cases are considered: digits, hyphens, punctuation marks, and the case of the letters (low and upper case). Generally, treating of digits involves an advanced lexical processing where number normalization is presented to unify formats. Hyphens pose another difficult decision to the lexical analyzer. Breaking up hyphenated words might be helpful in dealing with inconsistency of usage. For instance, this allows treating "state-of-the-art" and "state of the art" identically. However, there are words which include hyphens as an integral part. For instance, gilt-edge, B-49, etc. The most suitable procedure seems to adopt a general rule and specify the exceptions on a case basis.

  Normally, punctuation marks are removed entirely in the process of lexical analysis. While some punctuation marks are an integral part of the word, removing them does not seem to have an impact on the searching terms because the risk of misinterpretation in this case is minimal. However, very particular scenarios might require the preparation of a list of exceptions.

– *Elimination of stopwords*. A stopword can be a word without meaning in a specific language, or it can be a token that does not have linguistic meaning. The examples of stopwords in the English language are "a", "the", "is", etc. It is considered that the stopwords can be words which are too frequent among the documents in the collection because such words are not good discriminations. In fact, a word which occurs in 80% of the documents in the collection is useless for purposes of text understanding. Normally, stopwords are filtered out as important terms. Articles, prepositions, and conjunctions are natural candidates for a list of stopwords.

  Elimination of stopwords has an additional important benefit. It reduces the number of terms considerably. Since stopwords elimination also permits the reduction of the number of terms, the list of stopwords might be extended to include words other than articles, and conjunctions. For instance, some verbs, adverbs, and adjectives could be treated as stopwords.

  Despite these benefits, elimination of stopwords might reduce recall. For instance, consider a user who is trying to understand a document containing the phrase "to be or not to be". Elimination of stopwords might leave only the term *be* making it almost impossible to properly understand the documents which contain the specified phrase.

– *Stemming*. Frequently, the same word can be presented in a document in different syntactic variations. But only one variant of this word is relevant for the representation of the document. Plurals, gerund forms, and past tense suffixes are examples of syntactic variations which prevent a perfect match between words of the

document. This problem can be partially overcome with the substitution of the words by their respective stems.

A stem is the portion of a word which is left after the removal of its affixes. A typical example of a stem is the word "detect" which is the stem for the variants "detected", "detecting", "detection", and "detections". Stems are thought to be useful for improving searching of terms because they reduce variants of the same root word to a common concept. Furthermore, stemming has the secondary effect of reducing the number of terms used for the representation of documents because the number of distinct words is reduced.

The author of [17] distinguishes four types of stemming strategies: affix removal, table lookup, successor variety, and n-grams. Table lookup consists simply of looking for the stem of a word in a table. It is a simple procedure but requires considerable storage space. This type of stemming algorithm might not be practical. Successor variety stemming is based on the determination of morpheme boundaries, uses knowledge from structural linguistics, and is more complex than affix removal stemming algorithm. N-grams stemming is based on the identification of bigrams and trigrams and is more a term clustering procedure than a stemming one.

Affix removal stemming is intuitive, simple, and can be implemented efficiently. In affix removal, the most important part is suffix removal because most variants of a word are generated by the introduction of suffixes (instead of prefixes). While there are three of four well-known suffix removal algorithms, the most popular one is that proposed by Porter [18] because of its simplicity and elegance. Despite its simplicity, the Porter algorithm yields results comparable to those of the more sophisticated algorithms.

4 General Scheme of the Proposed Method

The general scheme of the proposed method consists of five steps:

Preprocessing. We test three configuration of preprocessing:

– Lexical analysis:

1. Digits – normalized, a point between numbers was detected.
2. Hyphens – considered as different words.
3. Punctuation marks – only for detection of sentences break, in other cases are eliminated.
4. The case of the letters – only for detection of sentence break, in other cases are eliminated.

– Combination of elimination of stopwords and stemming:

1. Stopwords are excluded from MFSs.
2. Stopwords are excluded from MFSs and stemming is applied. We used the stemmer algorithm of Porter (see [18] for details).
3. Stemming is used but stopwords are kept.

Note that the extracted sentences for the final summary remains without changes.

Maximal Frequent Sequences. MFSs are consecutive word sequences which appear in the text at least β times and are not contained in other frequent sequences. García [19] proposed an efficient algorithm to extract MFSs from a single document. In his works [19, 20] MFSs are formally defined. Let us give here only a short example. For the following text:

In Cholula you can find 365 churches and *the most beautiful is* of the pyramid.
The church of Tonantzintla is the most beautiful.
Xcaret is the most beautiful natural park of Cancun.
Cantona is the most beautiful city of the ancient cities.

resulted MFS for $\beta = 3$ is "*is the most beautiful*", while the only MFS for $\beta = 4$ is "*the most beautiful*" (it is not an MFS with $\beta = 3$ since it is not maximal with this β).

In this work, we used the algorithm to extract MFSs from [19]. We use $\beta = 2$, 3 and 4 in all experiments. The choice of β was discussed in [1].

Term Selection. When we say the terms, we referred to the features which we use in this step. The terms are MFSs extracted from a document which we denote as *MFS*. Also, in this paper we consider the words derived from MFSs which we denote as *W-MFS*.

Term Weighting. We test a scheme for weighting MFSs which take into account frequency of MFS, length of MFS, and frequency of derived terms from MFS. The terms T_i can be weighted in different manners, and have a weight t_i. This general scheme is defined as $p_i(t_j) = X \cdot Y$, where $p_i(t_j)$ - term weighting j in the documents i, X and Y can be determined as frequency of MFS, length of MFS, and frequency of derived terms from MFS. This term weighting scheme permits to detect which of the characteristics of MFS helps better to summarize a text.

Sentence Weighting. Sentence S_i has weight s_j = sum w_{ij}, contribution of T_i in S_j is $w_{ij} = f_{ij}$ x t_i, where f is a presence of T_i in D_j, t is an importance of T_i. Here f is 0 or 1.

Sentence Selection. This procedure completes a summary by adding the densest sentences or choosing the position of a sentence in a text until the summary is limited by the number of words. For the first option, we chose the sentences which have more weighting score. This type of methods is dominion-independent and can be applied for a variety of texts. For the second option, there are methods which are position-dependent and can be applied only for special topics (for example, news reports).

5 Results

Test data set. The proposed methods were tested with a collection Document Understanding Conference (DUC) provided in [21]. In particular, we use the data set of 567 news articles of different length and with different topics. For each document different summaries are given. Each summary which is used for evaluation was generated by two different experts manually. The experts were asked to generate summaries expressing the content of an original text by concatenating representative sentences.

Each expert was asked to generate summary of different length. We use summaries which have 100 words.

Evaluation. ROUGE evaluation toolkit [22] was used for evaluation. ROUGE is a method that is based on *n*-gram statistics, found to be highly correlated with human evaluations. This method compares the summaries made manually and generated by proposed methods. The following measures are used for evaluation. The evaluation is done using *n*-gram (1, 1) setting of ROUGE, which was found to have the highest correlation with human judgments, at a confidence level of 95%.

Baseline. The baseline configuration selects the first sentences of the text until the desired size is reached [21]. This configuration gives very good results on the kind of the texts (news reports) that we experimented with, but would not give so good results on other types of texts.

Preprocessing. For the first experimental procedure we extract MFSs without stop-words. Then, for the second experimental procedure we extract MFSs without stop-words and stemmed. Finally, for the third experimental procedure, MFS are only stemmed. In all experiments, we use stemming from [18] before extracting MFS.

Results with MFS. The results obtained in [2] are shown in Table 1. These results are shown for comparison with results obtained in Tables 2-4. Note that we show not all but only more relevant results for observing the effect of preprocessing on extractive summarization with MFSs.

Term Selection. First, MFSs *(MFS)* are selected for each document. Second, the terms derived from MFSs are extracted. The resulted terms derived from MFSs are words or *W-MFSs*.

Table 1. Results for comparison (plain MFSs)

Experimental configuration			Results		
Term Selection	Term Weighting	Sentence Selection	Recall	Precision	F-measure
MFS	*X·Y*	N_{sent}	0.43353	0.44737	0.44022
MFS	1	N_{sent}	0.44128	0.45609	0.44840
MFS	*X*	N_{sent}	0.43977	0.45587	0.44752
MFS	*Y*	N_{sent}	0.44085	0.45564	0.44796
W–MFS	1	N_{sent}	0.38364	0.40277	0.39284
W–MFS	*Z*	N_{sent}	0.44609	0.45953	0.45259
W–MFS	*Z*	$N_{com(s1)}$	**0.46576**	**0.48278**	**0.47399**
W–MFS	*Z*	$N_{com(s1+s2)}$	0.46158	0.47682	0.46895
MFS	*Y*	$N_{com(s1)}$	0.46381	0.48072	0.47185
MFS	1	$N_{com(s1)}$	0.46354	0.48072	0.47185

Term Weighting. First, the frequency of terms derived from MFSs is calculated for each document. Second, three weighting schemes described below are tested for selection of terms.

- First scheme: $p_i(t_j) = 1$, where $p_i(t_j)$ is a term weighting j in the documents i.
- Second scheme: $p_i(t_j) = X \cdot Y$, where X is a length of MFS, Y is a frequency of MFS. In Table 3, you can find the results for different combinations of X and Y.
- Third scheme: $p_i(t_j) = Z$, where Z is a frequency of MFS in a text.

Sentence Weighting. For each experiment, the sum of all term weights is calculated for each sentence.

Sentence Selection. We tested two sentence selection schemes:

- First scheme: sentences N_{sent} which have weight are composed a summary.
- Second scheme: the combination of sentences from different methods is used. We start composing a summary with the first sentence $N_{com(s1)}$ and the second sentence $N_{com(s1+s2)}$ obtained from the corresponding scheme, and then each summary is completed with some sentences from baseline until 100 words is reached. In Table 2 the number of sentences is indicated.

Experiment 1. In this experiment, we tried different term selection, term weighting and sentence selection schemes. Previously, the preprocessing for extraction of MFSs was done. The results of experimental procedure are presented in Table 2. The best results are highlighted with bold type of letter. We detect that the weighting scheme of frequency of words derived from MFSs gives the best sentence for a summary, and together with sentences obtained with baseline configuration, the best summary is obtained. For this experiment, we extract MFSs excluding stopwords.

Table 2. Results for experiment 1 (stopwords excluded)

Experimental configuration			Results		
Term Selection	Term Weighting	Sentence Selection	Recall	Precision	F-measure
MFS	$X \cdot Y$	N_{sent}	0.42689	0.43347	0.43005
MFS	1	N_{sent}	0.44193	0.44426	0.44298
MFS	X	N_{sent}	0.42263	0.42961	0.42599
MFS	Y	N_{sent}	0.44678	0.44849	0.44752
W–MFS	1	N_{sent}	0.39657	0.39834	0.39733
W–MFS	Z	N_{sent}	0.45504	0.45626	0.45553
W–MFS	Z	$N_{com(s1)}$	0.46416	0.48090	0.47226
W–MFS	Z	$N_{com(s1+s2)}$	0.46033	0.47532	0.46759
MFS	Y	$N_{com(s1)}$	0.44605	0.44771	0.44676
MFS	1	$N_{com(s1)}$	**0.46266**	**0.47979**	**0.47094**

Experiment 2. For this experiment, we change preprocessing configuration: MFS are stemmed and stopwords are excluded from MFSs. See results in Table 3. We observed that stopwords slightly influence on results that stemming because, as in previous experiment, the presence of MFSs influenced on obtaining the best summary.

Table 3. Results for experiment 2 (stopwords excluded + stemming)

Experimental configuration			Results		
Term Selection	Term Weighting	Sentence Selection	Recall	Precision	Fmeasure
MFS	*X·Y*	N_{sent}	0.42538	0.43151	0.42831
MFS	1	N_{sent}	0.44315	0.44517	0.44405
MFS	*X*	N_{sent}	0.41837	0.42496	0.42153
MFS	*Y*	N_{sent}	0.44538	0.44681	0.44598
W–MFS	1	N_{sent}	0.39657	0.39834	0.39733
W–MFS	*Z*	N_{sent}	0.45576	0.45679	0.45615
W–MFS	*Z*	$N_{com(s1)}$	0.46413	0.48081	0.47220
W–MFS	*Z*	$N_{com(s1+s2)}$	0.46259	0.47721	0.46966
MFS	*Y*	$N_{com(s1)}$	0.46432	0.48139	0.47258
MFS	1	$N_{com(s1)}$	**0.46456**	**0.48169**	**0.47285**

Experiment 3. For this experiment, we change preprocessing configuration: MFSs are stemmed and stopwords are included. The results are shown in Table 4. We confirmed that the results are changed because of the presence of stopwords: in this experiment the summary is obtained weighting the frequency of MFSs.

Table 4. Results for experiment 3 (stemming but stopwords kept)

Experimental configuration			Results		
Term Selection	Term Weighting	Sentence Selection	Recall	Precision	Fmeasure
MFS	1	N_{sent}	0.43971	0.44234	0.44067
MFS	*X*	N_{sent}	0.43380	0.43664	0.43487
MFS	*Y*	N_{sent}	0.43867	0.44100	0.43949
MFS	1	N_{sent}	0.39657	0.39834	0.39733
W–MFS	*Z*	N_{sent}	0.44609	0.44632	0.44608
W–MFS	*Z*	$N_{com(s1)}$	0.46486	0.48189	0.47310
W–MFS	*Z*	$N_{com(s1+s2)}$	0.46293	0.47831	0.47037
W–MFS	*Y*	$N_{com(s1)}$	**0.46508**	**0.48233**	**0.47343**
MFS	1	$N_{com(s1)}$	0.46461	0.48182	0.47293

Comparison. We compare with the state-of-the-art methods like *TextRank* [12]. Specifically, *DirectedBackward* version of *TextRank* was evaluated in the same conditions as proposed methods (see Table 5, *TextRank*) and the same version of *TextRank* with preprocessing (see Table 5, *TextRank w/preprocessing*). And also we compare results presented in [2] (see Table 5, *MFS w/o preprocessing*). Finally, the best version of each experimental procedure of this paper is included (see *MFS w/ preprocessing* 1, 2, and 3).

Table 5. Results with other methods

Method	Recall	Precision	F-measure
TextRank	0.45220	0.43487	0.44320
TextRank w/preprocessing	0.46582	0.48382	0.47450
MFS w/o preprocessing	**0.46576**	**0.48278**	**0.47399**
MFS w/preprocessing 1	0.46266	0.47979	0.47094
MFS w/preprocessing 2	0.46456	0.48169	0.47285
MFS w/preprocessing 3	0.46508	0.48233	0.47343

We can see that preprocessing does not affect positively obtaining terms for extractive summarization, at least not in the case of MFSs.

6 Conclusions

We modified our automatic single-document text summarization method based on MFSs as terms by including preprocessing stage. We found, however, that preprocessing does not affect positively the summaries obtained with our method. This is good news and bad news. Bad because we did not find better terms, and our summaries did not improve. Good because we confirmed that classic plain MFSs (sequences of wordforms and not stems or only significant words), which are calculated in a totally language-independent manner, are good terms for this task.

On the other hand, since we showed that our preprocessing almost does not either affect the results negatively, one can exclude stopwords and word endings from processing and still obtain almost the same quality of extractive summarization. Excluding stopwords significantly reduces the risk of exponential explosion of the size of the data structures used to mine for MFSs and for their application in our method, as well as the number of the terms (MFSs or ngrams) dealt with.

In our future work we expect to see how preprocessing affects multi-document (as opposed to single document considered here) summarization.

Acknowledgments. This work was done under the partial support of Mexican Government (CONACyT, SNI, SIP-IPN). We also thank CIC-IPN and UAEM for their assistance.

References

1. Ledeneva, Y., Gelbukh, A., García-Hernández, R.: Terms Derived from Frequent Sequences for Extractive Text Summarization. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 593–604. Springer, Heidelberg (2008)
2. Ledeneva, Y., Gelbukh, A., García-Hernández, R.: Keeping Maximal Frequent Sequences Facilitates Extractive Summarization. In: Sidorov, G., et al. (eds.) Advances in Computer Science and Engineering, 9th Conference on Computing (CORE-2008), Research in Computing Science, vol. 34, pp. 163–174 (2008) ISSN: 1870-4069
3. Pomikálek, J., Rehurek, R.: The Influence of preprocessing parameters on text categorization. In: Proc. of World Academy of Science, Engineering and Technology, vol. 21, pp. 430–434 (2007)

4. Abu-Salem, H., Al-Omari, M., Evens, M.W.: Stemming methodologies over individual words for an Arabic Information Retrieval System. Journal of the American Society for Information Science 50, 524–529 (1999)
5. Larkey, L.S., Ballesteros, L., Connell, M.: Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. In: Proc. of ACM SIGID Conference in IR, pp. 275–282 (2002)
6. Halácsy, P., Trón, V.: Benefits of Resource-Based Stemming in Hungarian Information Retrieval. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 99–106. Springer, Heidelberg (2007)
7. Hamzah, M.P., Tengku Sembok, M.: On Retrieval Performance of Malay Textual Documents. In: Proc. of IASTED, pp. 156–161. ACTA Press (2006)
8. Frakes, W., Baeza-Yates, R.: Information Retrieval: Data Structures and Algorithms. Prentice Hall, Englewood Cliffs (1992)
9. Villatoro-Tello, E., Villaseñor-Pineda, L., Montes-y-Gómez, M.: Using Word Sequences for Text Summarization. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 293–300. Springer, Heidelberg (2006)
10. Liu, D., et al.: Multi-Document Summarization Based on BE-Vector Clustering. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 470–479. Springer, Heidelberg (2006)
11. Bolshakov, I.A.: Getting One's First Million...Collocations. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 229–242. Springer, Heidelberg (2004)
12. Sidorov, G., Gelbukh, A.: Automatic Detection of Semantically Primitive Words Using Their Reachability in an Explanatory Dictionary. In: IEEE International Workshop on Natural Language Processing and Knowledge Engineering, NLPKE 2001 at Proc. International IEEE SMC-2001 Conference: Systems, Man, And Cybernetics, USA, pp. 1683–1687 (2001) ISBN 0-7803-7087-2
13. Song, Y., et al.: A Term Weighting Method based on Lexical Chain for Automatic Summarization. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 636–639. Springer, Heidelberg (2004)
14. Mihalcea, R.: Random Walks on Text Structures. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 249–262. Springer, Heidelberg (2006)
15. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain (2004)
16. Baeza-Yates, R.: Modern Information Retrieval. Addison Wesley/Longman Publishing Co. (1999)
17. Frakes, W., Baeza-Yates, R.: Information Retrieval: Data Structures and Algorithms. Prentice-Hall, Englewood Cliffs (1992)
18. Sparck Jones, K., Willet, P.: Readings in Information Retrieval. Morgan Kaufmann, San Francisco (1997)
19. García-Hernández, R.A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: A Fast Algorithm to Find All the Maximal Frequent Sequences in a Text. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) CIARP 2004. LNCS, vol. 3287, pp. 478–486. Springer, Heidelberg (2004)
20. García-Hernández, R.A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: A New Algorithm for Fast Discovery of Maximal Sequential Patterns in a Document Collection. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 514–523. Springer, Heidelberg (2006)
21. DUC. Document understanding conference 2002 (2002), http://www-nlpir.nist.gov/projects/duc
22. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proc. of Workshop on Text Summarization of ACL, Spain (2004)

Text Summarization by Sentence Extraction Using Unsupervised Learning*

René Arnulfo García-Hernández, Romyna Montiel, Yulia Ledeneva, Eréndira Rendón, Alexander Gelbukh, and Rafael Cruz

Pattern Recognition Laboratory, Toluca Institute of Technology, Mexico
Autonomous University of the State of Mexico, Mexico
Center for Computing Research, National Polytechnic Institute, Mexico
SoNet RC, University of Center Europe in Skalica, Slovakia
renearnulfo@hotmail.com, romyna.montiel@yahoo.com.mx,
yledeneva@yahoo.com, www.Gelbukh.com

Abstract. The main problem for generating an extractive automatic text summary is to detect the most relevant information in the source document. Although, some approaches claim being domain and language independent, they use high dependence knowledge like key-phrases or golden samples for machine-learning approaches. In this work, we propose a language- and domain-independent automatic text summarization approach by sentence extraction using an unsupervised learning algorithm. Our hypothesis is that an unsupervised algorithm can help for clustering similar ideas (sentences). Then, for composing the summary, the most representative sentence is selected from each cluster. Several experiments in the standard DUC-2002 collection show that the proposed method obtains more favorable results than other approaches.

1 Introduction

In the last two decades, we have experienced an exponential increase in the electronic text information available for being query. The best example of the hugest and ever-increased collection of documents most frequently querying is Internet, with millions of web documents. Nowadays, it is common to use Google for retrieving a list of thousands of web pages, but the user has to decide if a document is interesting only with the extracted text where the words of the request query appears. Therefore, it is necessary download and read each document until the user finds satisfactory information. It was unnecessary and time-consuming routine. Thus, it is indispensable to develop automatic methods for detecting the most relevant content from a source document in order to show it as a summary. In addition, there are a number of scenarios where automatic construction of such summaries is useful. Other examples include automatic construction of summaries of news articles or email messages for sending them to mobile devices as SMS; summarization of information for government officials, businesspersons, researches, etc., and summarization of web pages to

* Work done under partial support of Mexican Government: CONACyT, SNI, SIP-IPN, PIFI-IPN.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 133–143, 2008.

be shown on the screen of a mobile device, among many others. These examples show that it is desirable that text summarization approaches work more in language and dominion independent way.

Automatic Text Summarization (ATS) is an active research area that deals with single- and multi-document summarization tasks. In single-document summarization, the summary of only one document is built, while in multi-document summarization the summary of a whole collection of documents (such as all today's news or all search results for a query) is built. While we believe that our ideas apply to either case, in this work we have experimented only with single-document summaries.

Summarization methods can be classified into abstractive and extractive summarization [1]. An abstractive summary is an arbitrary text that describes the contexts of the source document. Abstractive summarization process consists of "understanding" the original text and "re-telling" it in fewer words. Namely, an abstractive summarization method uses linguistic methods to examine and interpret the text and then to find new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original document. While this may seem the best way to construct a summary (and this is how human beings do it), in real-life setting immaturity of the corresponding linguistic technology for text analysis and generation currently renders such methods practically infeasible.

An extractive summary, in contrast, is composed with a selection of sentences (or phrases, paragraphs, etc.) from the original text, usually presented to the user in the same order—i.e., a copy of the source text with most sentences omitted. An extractive summarization method only decides, for each sentence, whether or not it will be included in the summary. The resulting summary reads rather awkward; however, simplicity of the underlying statistical techniques makes extractive summarization an attractive, robust, language-independent alternative to more "intelligent" abstractive methods. In this paper, we consider extractive summarization.

A typical extractive summarization method consists in several steps, at each of them different options can be chosen. We will assume that the units of selection are sentences (these could be, say, phrases or paragraphs). Thus, final goal of the extractive summarization process is sentence selection.

The main problem for generating an extractive automatic text summary is to detect the most relevant information in the source document. Although, some approaches claim being domain and language independent, they use some degree of language knowledge like lexical information [2], key-phrases [3] or golden samples for supervised learning approaches [4-6]. Furthermore, training on a specific domain tends to customize the extraction process to that domain, so the resulting classifier is not necessarily portable. In our opinion, these works present a high dominion and language dependence degree.

In this work, we propose a language- and domain-independent automatic text summarization approach by sentence extraction using an unsupervised learning algorithm. Our hypothesis is that an unsupervised algorithm can help for clustering similar ideas (sentences). Then, for composing the summary, the most representative sentence is selected from each cluster. In addition, this approach lets to control in some degree the number of words in the generated summary.

The paper is organized as follows. Section 2 summarizes the state-of-the-art of text summarization methods. Section 3 describes the general scheme of the proposed

approach. Section 4 presents the experimental settings followed for the experimentation. Section 5 compares our approach with those of existing methods. Section 6 concludes the paper.

2 Related Work

Ledeneva *et al.* [7, 8] suggest a typical automatic extractive summarization approach composed by term selection, term weighting, sentence weighting and sentence selection steps. One of the ways to select the appropriate sentences is to assign some numerical measure of usefulness of a sentence for the summary and then select the best ones; the process of assigning these usefulness weights is called *sentence weighting*. One of the ways to estimate the usefulness of a sentence is to sum up usefulness weights of individual terms of which the sentence consists; the process of estimating the individual terms is called *term weighting*. For this, one should decide what the terms are: for example, they can be words; deciding what objects will count as terms is the task of *term selection*. Different extractive summarization methods can be characterized by how they perform these tasks.

Ideally, a text summarization system should "understand" (analyze) the text and express its main contents by generating the text of the summary. For example, Cristea *et al.* [9] perform sentence weighting according to their proximity to the central idea of the text, which is determined by analysis of the discourse structure.

However, the techniques that try to analyze the structure of the text involve too sophisticated and expensive linguistic processing. In contrast, most of the methods discussed in the literature nowadays represent the text and its sentences as a bag of simple features, using statistical processing without any attempts to "understand" the text.

Supervised learning methods consider sentence selection as classification: they train a classifier using a collection of documents supplied with existing summaries. As features of a sentence, such methods can consider text units (in which case we can speak of term selection) or other, non-lexical characteristics. Villatoro-Tello *et al.* [4] use as terms *n*-grams found in the text. Kupiec *et al.* [10] use predefined cue phrases (this makes the method language- and domain-dependent) as well as non-lexical features such as the position and length of the sentence; their sentence weighting procedure also includes measuring the overlap of the sentence with the title of the document. HaCohen-Kerner *et al.* [3] consider many other lexical and non-lexical features, such as the position of the sentence in the paragraph.

However, the majority of current methods are purely heuristic: they do not use any learning but directly state the procedure used for term selection, term weighting, and/or sentence weighting (given that sentence selection in most cases consists in selecting the best-weighted sentences).

A very old and very simple sentence weighting heuristic does not involve any terms at all: it assigns highest weight to the first sentences of the text. Texts of some genres—such as news reports or scientific papers—are specifically designed for this heuristic: e.g., any scientific paper contains a ready summary at the beginning. This gives a baseline [11] that proves to be very hard to beat on such texts. It is worth noting that in Document Understanding Conference (DUC) competitions [11] only five systems performed above this baseline, which does not demerit the other systems because this baseline is genre-specific.

Of the works devoted to term-based methods, most concentrate on term weighting. Xu *et al*. [12] derives relevance of a term from an ontology constructed with formal concept analysis. Song *et al*. [2] basically weight a word basing on the number of lexical connections, such as semantic associations expressed in a thesaurus, that the word has with its neighboring words; along with this, more frequent words are weighted higher. Mihalcea [13, 14] presents a similar idea in the form of a neat, clear graph-based formalism: the words that have closer relationships with a greater number of "important" words become more important themselves, the importance being determined in a recursive way similar to the PageRank algorithm used by Google to weight web pages.

The latter idea can be applied directly to sentence weighting without term weighting: a sentence is important if it is related to many important sentences, where relatedness can be understood as, say, overlap of the lexical contents of the sentences [13].

Recently, a novel approach quite different from other methods was presented by Ledeneva *et al.* [7]. In this work, the sentences are weighted by using the terms derived from the maximal frequent word sequences. Then, the best sentence is combined with the baseline sentences for composing the summary. This approach is ranked, according to ROGUE evaluation system, in third place.

The methods presented in [13-15] and [7] are those that currently give the best results and with which we compare our suggested method.

While in the experiments reported in the papers discussed above were based on words as terms, this is not the only possible option. Liu *et al*. [16] uses pairs of syntactically connected words (basic elements) as atomic features (terms). Such pairs (which can be thought of as arcs in the syntactic dependency tree of the sentence) have been shown to be more precise semantic units than words [17, 18, 19]. However, while we believe that trying text units larger than a word is a good idea, extracting the basic elements from the text requires dependency syntactic parsing, which is language-dependent. Simpler statistical methods, as the use of *n*-grams as terms in [4], may prove to be more robust and language-independent.

3 General Scheme of the Proposed Approach

Usually an extractive summarization approach performs term selection, term weighting, sentence weighting, and sentence selection steps. However, the strategy of sentence selection step is reduced to simply taking the sentences with highest weight. Even though this strategy works well for the first ranked sentence, the strategy could lead for sentences similar to the first one to tend to be ranked after the first one; producing redundant sentences in the summary. This problem affects negatively in recall measure. In this sense, we propose to substitute the sentence weighting and sentence selection steps, with an unsupervised learning algorithm. Our hypothesis is that an unsupervised learning algorithm could help for automatically detecting the groups of similar sentences from which is selected the most representative sentence; reducing in this way the redundancy in the summary. In this section, we describe the general steps that are followed in the proposed approach.

3.1 Term Selection

An *n*-gram is a sequence of *n* words. We say that an *n*-gram occurs in a text if these words appear in the text in the same order immediately one after another, e.g., a 4-gram (*n*-gram of length 4) *words appear in the text* occurs once in the previous sentence, while *appear immediately after another* does not (these words do not appear on adjusting positions), neither does *the text appear in* (order is different).

The definition of n-gram depends on what one considers words. For example, one can consider capitalized (*Mr. Smith*) and non-capitalized (*a smith*) words as the same word or as different words; one can consider words with the same morphological stem (*ask*, *asked*, *asking*), the same root (*derive*, *derivation*), or the same meaning (*occur*, *appear*) as the same word; one can omit the stop-words (*the*, *in*) when counting word positions, etc. Say, one can consider that in our example sentence above there occur the *n*-grams *we say* (capitalization ignored), *word appear* (plural ignored), *appear text* (*in the* ignored). This can affect counting the *n*-grams: if one considers *occur* and *appear* as equivalent and ignores the stop-words, then in our example sentence the bigram *appear text* occurs twice.

3.2 Term Weighting

Boolean Weighting (BOOL): It is the easiest way to weight a term. It models the presence or absence of a term in the document, defined as:

$$w_i(t_j) = \begin{cases} 1, & \text{if the term } t_j \text{ appears in document } i \\ 0, & \text{otherwise} \end{cases}$$

Term Frequency (TF) was proposed in [20]. This weighting takes into account that a term that occurs in a document can better reflect the contents of document than a term that occurs less frequent. Therefore, the weighting TF assigns a greater relevance to terms with greater frequency and consists in evaluating the number of times the term appears in the document.

$w_i(t_j) = f_{ij}$, where f_{ij} is the frequency of the term j in document i.

Inverse Document Frequency (IDF) was proposed by Salton [21] for improving information retrieval systems (IR). The problem of TF weighting in IR is that, when a term appears in almost all the documents in the collection; this term is useless for discriminating relevant documents. For example, the stop-word *and* could have a high TF, but it is useless for discriminating the relevant documents since tends to appear in most of the documents. IDF is defined as:

$$w_i(t_j) = \log\left(\frac{N}{n_j}\right),$$

where N is the number of documents in the collection and n_j is the number of documents where the term j appears.

TF-IDF. The problem of IDF weighting in IR is that it is not possible distinguish between two documents with the same vocabulary (list of different words), even

thought if the term is more frequent in a document. TF-IDF weighting gives more relevance to the terms that are less frequent in the collection but more frequent into the document.

$$w_i(t_j) = f_{ij} \times \log\left(\frac{N}{n_j}\right)$$

Note that in this paper we propose to use these term weights for single document summarization. Therefore, for applying these term weights we can consider the document as a collection of sentences instead of a collection of documents.

3.3 Sentence Selection Using an Unsupervised Learning

An unsupervised learning algorithm form groups of objects in order to achieve, in the one hand, the greatest possible similarity between objects of a group, in other hand, the greatest possible dissimilarity between objects of different groups.

In this step, we propose to use an unsupervised algorithm for discovering the groups of sentences with similar meaning. Then, we can select the most representative sentence from each group in order to compose the summary. In particular, we propose to use the well-known K-means algorithm, which assumes that the number of clusters is previously known. Sometimes this characteristic is a disadvantage in K-means, however in our proposed approach is an advantage because let to specify the number of groups to create what allowed, at the same time, to estimate the number of words in the final summary. For example, if the average of words per sentence is 20 and a user desires a 100-word summary then K-means must create 5 clusters, obviously this is only an estimation of the number of words in the final summary. K-means represents each sentence in a vector space model. So, each document is represented as a vector of features, where the features correspond to the different terms in the document, in this case *n*-grams.

K-means is based on centroids, which are points in the vector space model calculated as the mean of the objects in a group. K-means iteration consists in to assign each object to the closest centroid and then the new centroids are recalculated again. The algorithm finishes when the centroids do not change. In the beginning, the K-means algorithm need seeds as the initial centroids for each group. Thus, the successful of K-means depends on selecting good initial seeds. Normally, the initial seeds are selected in random way. In our proposed approach, first sentences are considered as initial seeds, since it is known that the Baseline sentences are good candidate sentences for composing the summary. For measuring the similarity between two sentences the Euclidean distance is used, defined as:

$$\text{Distance}(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2},$$

where X and Y are sentences expressed as vectors with n features.

4 Experimental Results

We have conducted several experiments to verify our hypotheses formulated in the previous section.

Algorithm: In each experiment, we followed the standard sequence of steps:

- *Preprocessing*: eliminate stop-words, then apply Porter stemming [24];
- *Term selection*: decide which size of *n*-grams as features are to be used to describe the sentences;
- *Term weighting*: decide how the importance of each feature is to be calculated, it can be BOOL, TF, IDF or TFIDF;
- *Sentence clustering*: decide the initial seeds for the K-means algorithm, in this case Baseline sentences;
- *Sentence selection*: after K-means finishes, select the closest sentence (most representative sentence) to each centroid for composing the summary;
- The specific settings for each step varied between the experiments and are explained below for each experiment.

Test data set. We used the standard DUC 2002 collection [11]. In particular, we used the data set of 567 news articles of different length and with different topics. Each document in the DUC collection is supplied with a set of human-generated summaries provided by two different experts.[1] While each expert was asked to generate summaries of different length, we used only the 100-word variants.

Evaluation procedure. We used the ROUGE evaluation toolkit [22] which was found to highly correlate with human judgments [23]. It compares the summaries generated by the program with the human-generated (gold standard) summaries. For comparison, it uses *n*-gram statistics. Our evaluation was done using *n*-gram (1, 1) setting of ROUGE, which was found to have the highest correlation with human judgments, namely, at a confidence level of 95%. ROUGE calculates Precision, Recall, and F-measure values. We consider F-measure because it represents a balance (not an average) of Recall and Precision results.

Next tables show the results obtained with ROUGE for different gram size and different term weights. In the beginning, we only test for 1-grams to 5-grams, however, the results tends to be better as the size of *n*-gram was increased. Therefore, it was necessary to test until 11-gram. In particular, tables 1, 2 and 3 shows Recall, Precision and F-measure results, respectively. As F-measure shows, the worst result was obtained with 1-grams and BOOL; and the best result was obtained with 10-grams and IDF. In addition, these combinations were the worst and best results for recall and precision.

[1] While the experts were supposed to provide extractive summaries, we observed that the summaries provided in the collection were not strictly extractive: the experts considerably changed the sentences as compared with the original text.

Table 1. Recall on 100-word summaries for different sizes of *n*-grams and its weights

Term Selection	Term Weighting			
	BOOL	TF	IDF	TFIDF
1-grams	0.47517	0.47686	0.47632	0.47545
2-grams	0.47705	0.47694	0.47779	0.47777
3-grams	0.47940	0.47940	0.47932	0.47932
4-grams	0.47891	0.47891	0.47916	0.47913
5-grams	0.47942	0.47942	0.47910	0.47910
6-grams	0.47989	0.47979	0.48020	0.48020
7-grams	0.47976	0.47992	0.47964	0.47993
8-grams	0.48113	0.48072	0.48075	0.48055
9-grams	0.48084	0.48084	0.48020	0.48109
10-grams	0.48058	0.48103	**0.48155**	0.48101
11-grams	0.48004	0.47903	0.47856	0.47856

Table 2. Precision 100-word summaries for different sizes of *n*-grams and its weights

Term Selection	Term Weighting			
	BOOL	TF	IDF	TFIDF
1-grams	0.47039	0.47219	0.47168	0.47078
2-grams	0.47211	0.47204	0.47284	0.47284
3-grams	0.47452	0.47454	0.47441	0.47441
4-grams	0.47410	0.47410	0.47432	0.47429
5-grams	0.47462	0.47462	0.47432	0.47432
6-grams	0.47495	0.47497	0.47530	0.47530
7-grams	0.47493	0.47510	0.47487	0.47512
8-grams	0.47633	0.47606	0.47588	0.47587
9-grams	0.47632	0.47632	0.47553	0.47654
10-grams	0.47575	0.47609	**0.47684**	0.47634
11-grams	0.47529	0.47409	0.47370	0.47370

Table 3. F-measure on 100-word summaries for different sizes of *n*-grams and its weights

Term Selection	Term Weighting			
	BOOL	TF	IDF	TFIDF
1-grams	0.47264	0.47439	0.47387	0.47298
2-grams	0.47445	0.47436	0.47519	0.47517
3-grams	0.47683	0.47684	0.47673	0.47673
4-grams	0.47638	0.47638	0.47661	0.47658
5-grams	0.47689	0.47689	0.47658	0.47658
6-grams	0.47729	0.47725	0.47762	0.47762
7-grams	0.47721	0.47738	0.47713	0.47739
8-grams	0.47860	0.47826	0.47818	0.47808
9-grams	0.47845	0.47845	0.47773	0.47868
10-grams	0.47803	0.47842	**0.47906**	0.47854
11-grams	0.47753	0.47642	0.47599	0.47599

5 Comparison

We compared our proposed approach with the following results:

- State of the art 1: The author of [13] provided data, which were evaluated in the same conditions as proposed methods. Specifically, DirectedBackward version of TextRank [13] was evaluated. We also list the results of the original TextRank with implementation of PageRank with DirectedBackward version of TextRank but with some additional data processing to remove noisy data [14] and the modified TextRank with a biased version of PageRank [15]. See details of the preprocessing in [13–15].
- State of the art 2: We compare our methods with the best results reported in [7].
- Baseline (first): We denote *Baseline: first* the baseline mentioned in Section 2, which selects the first sentences of the document until the desired size of the summary is reached [7]. This Baseline gives very good results on the kind of documents (news reports) that we experimented with, but would not give so good results on other types of texts.
- Baseline (random): Ledeneva *et al.* [7] proposed another baseline heuristic, denoted *Baseline: random*, that is the average evaluation of selecting 10-times random sentences; we believe this to be a more realistic baseline for the types of texts other than news reports.

The comparison of the best F-measure results of our proposed approach with the above state-of-the-art approaches is presented in table 4. In this table, it is possible to observe that the difference between the worst and best approach is 0.09251. It difference was calculate in order to show that a centesimal or millesimal increase in F-measure is significant. The difference between Baseline *(first)* and Ledeneva [11] is 0.00105; and between Ledeneva [11] and TextRank [14] is 0.00051; however, the difference between TextRank [14] and our proposed approach is 0.00456. In addition, we calculate the difference between recall and precision in order to show how our proposed method gets the best balance between recall and precision. In addition, it is interesting to observe that, in comparison with Baseline (first), our approach obtained better recall results in all the experiments, while the precision was worse in all the experiments. Nevertheless, our proposed approach obtains better F-measure results than Baseline (first), except for 1-gram and BOOL weighting. It is good to mention the best recall result was obtained by our proposed approach.

Table 4. Results with other methods

Method	Recall	Precision	F-measure	|Recall-Precision|
Baseline: *random*	0.37892	0.39816	0.38817	0.01924
TextRank: [13]	0.45220	0.43487	0.44320	0.01733
Baseline: *first*	0.46407	0.48240	0.47294	0.01833
Ledeneva: [7]	0.46576	0.48278	0.47399	0.01702
TextRank: [14]	0.46582	0.48382	0.47450	0.018
Proposed	**0.48155**	**0.47633**	**0.47906**	**0.00522**
TextRank: [15]	0.47207	0.48990	0.48068	0.01783

6 Discussion and Conclusions

In this work, we proposed an extractive automatic text summarization approach by sentence extraction using an unsupervised learning algorithm. In particular, the K-means algorithm for creating groups of similar sentences was used. Then, from the groups of sentences, the most representative sentence was selected for composing the summary. Normally, the definition of the number of groups to form and the initial seeds of the groups are considered as disadvantages of K-means. However, these parameters are used to take advantage of Baseline sentences in order to improve the quality of the summaries. The proposed approach, in contrast to supervised methods, does not need large amount of golden samples for training. Therefore, our proposed approach is more independent from language and dominion.

According to experimental results we demonstrate that the proposed approach obtains more favorable results than others state-of-the-art approaches; ranking our proposed approach in second place, very close to the first place. In addition, our proposed approach outperforms the Baseline (*first*) heuristic for F-measure results, except for 1-gram and BOOL weighting.

References

1. Lin, C.Y., Hovy, E.: Automated Text Summarization in SUMMARIST. In: Proc. of ACL Workshop on Intelligent, Scalable Text Summarization, Madrid, Spain (1997)
2. Song, Y., et al.: A Term Weighting Method based on Lexical Chain for Automatic Summarization. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 636–639. Springer, Heidelberg (2004)
3. HaCohen-Kerner, Y., Zuriel, G., Asaf, M.: Automatic Extraction and Learning of Keyphrases from Scientific Articles. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 657–669. Springer, Heidelberg (2005)
4. Villatoro-Tello, E., Villaseñor-Pineda, L., Montes-y-Gómez, M.: Using Word Sequences for Text Summarization. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 293–300. Springer, Heidelberg (2006)
5. Chuang, T.W., Yang, J.: Text Summarization by Sentence Segment Extraction Using Machine Learning Algorithms. In: Proc. of the ACL 2004 Workshop, Barcelona, España (2004)
6. Neto, L., Freitas, A.A., Kaestner, C.A.A.: Automatic Text Summarization using a Machine learning Approach. In: Proceedings of the ACL 2004 Workshop, Barcelona, España (2004)
7. Ledeneva, Y., Gelbukh, A., García, H.R.: Terms Derived from Frequent Sequences for Extractive Text Summarization. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 593–604. Springer, Heidelberg (2008)
8. Ledeneva, Y., Gelbukh, A., García, H.R.: Keeping Maximal Frequent Sequences Facilitates Extractive Summarization. Research in Computing Science 34 (2008)
9. Cristea, D., Postolache, O., Pistol, I.: Summarization through Discourse Structure. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 632–644. Springer, Heidelberg (2005)
10. Kupiec, J., Pedersen, J.O., Chen, F.: A Trainable Document Summarizer. In: Proc. 18th ACM-SIGIR Conf. on Research and Development in Information Retrieval, pp. 68–73 (1995)

11. DUC. Document Understanding Conference 2002 (2002), http://www-nlpir.nist.gov/projects/duc
12. Xu, W., Li, W., Wu, M., Li, W., Yuan, C.: Deriving Event Relevance from the Ontology Constructed with Formal Concept Analysis. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 480–489. Springer, Heidelberg (2006)
13. Mihalcea, R.: Random Walks on Text Structures. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 249–262. Springer, Heidelberg (2006)
14. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proc. Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain (2004)
15. Hassan, S., Mihalcea, R., Banea, C.: Random-Walk Term Weighting for Improved Text Classification. In: Proc. Semantic Computing (ICSC 2007), Irvine, CA (2007)
16. Liu, D., He, Y., Ji, D., Hua, J.: Multi-Document Summarization Based on BE-Vector Clustering. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 470–479. Springer, Heidelberg (2006)
17. Bolshakov, I.A.: Getting One's First Million...Collocations. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 229–242. Springer, Heidelberg (2004)
18. Koster, C.H.A.: Transducing Text to Multiword Units. In: Workshop on Multiword Units MEMURA at 4th Int. Conf. on Language Resources and Evaluation, LREC 2004, Portugal (2004)
19. Sidorov, G., Gelbukh, A.: Automatic Detection of Semantically Primitive Words Using Their Reachability in an Explanatory Dictionary. In: Proc. Int. Workshop on Natural Language Processing and Knowledge Engineering, NLPKE 2001, USA, pp. 1683–1687 (2001)
20. Luhn, H.P.: A Statical Approach to Mechanical Encoding and Searching of Literary Information. IBM Journal of Research and Development, 309–317 (1975)
21. Salton, G., Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. Information Processing & Management 24, 513–523 (1988)
22. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of Workshop on Text Summarization of ACL, Spain (2004)
23. Lin, C.Y., Hovy, E.: Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In: Proceedings of HLT-NAACL, Canada (2003)
24. Spark Jones, K., Willet, P.: Reading in Information Retrieval. Morgan Kaufmann, San Francisco (1997)

Spanish Nested Named Entity Recognition Using a Syntax-Dependent Tree Traversal-Based Strategy

Yunior Ramírez-Cruz and Aurora Pons-Porrata

Center for Pattern Recognition and Data Mining
Universidad de Oriente
Santiago de Cuba, Cuba, 90500
{yunior,aurora}@cerpamid.co.cu

Abstract. In this paper, we address the problem of nested Named Entity Recognition (NER) for Spanish. Phrase syntactic structure is exploited to generate a tree representation for the set of phrases that are candidate to be named entities. The classification of all candidate phrases is treated as a single problem, for which a globally optimal solution is approximated using a strategy based on the postorder traversal of that representation. Experimental results, obtained in the framework of SemEval 2007 Task 9 NER subtask, demonstrate the validity of our approach.

1 Introduction

Named Entity Recognition (NER) is a basic step for a number of tasks such as Information Extraction, Question Answering and Automatic Summarization. The classic NER task was introduced in the Message Understanding Conferences and consists in detecting and classifying elemental information units contained in text documents such as proper names (persons, organizations, locations, etc.), quantities and temporal expressions. The classic problem definition considers no nesting or overlapping between different Named Entities (NE's).

Extensive work has been conducted on NER. The greatest effort has been focused on English documents covering the biomedical and newswire domains. While the annotation of benchmark corpora for English NER in the biomedical domain considers nested NE's, this is not common for the newswire domain. Because of this, research efforts to deal with the problem of recognizing nested structures have been largely confined to the former domain, although nested NE's are likely to occur in any knowledge domain.

Until recently, Spanish corpora have often lacked the annotation of nested NE's. Consequently, a small number of works have addressed the problem for that language. However, being able to recognize all NE's is crucial for other tasks depending on it, such as coreference resolution and scenario template matching, since nested structures implicitly contain relations that may help improve their performance.

In this paper, we focus on the nested NER problem for the Spanish language. Unlike most common approaches, which treat the problem either as a postprocessing stage of the classic NER problem or as a combination of several instances

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 144–154, 2008.

of it, we address the recognition of all NE's included in a nested structure as a single problem. For each sentence, phrases that are candidate to be NE's are detected in the deep constituency tree. A second tree containing the representations of all candidate phrases through a set of boolean features is generated. The structural and functional relations imposed to the candidate phrases by the syntax are encoded in that tree. Finally, a globally optimal classification for all candidate phrases is approximated using a strategy based on the postorder traversal of the representation tree.

For evaluation purposes, we use the SemEval 2007 Task 9 Spanish dataset for the NER subtask. We evaluate the impact of several elements in our model and establish a comparison between the results obtained by our method and those reported for that subtask.

The rest of the paper is structured as follows. In Section 2 we review previous work on nested NER. Section 3 is devoted to describing our approach, whereas Section 4 contains the description of the experiments that we carried out. Finally, we expose our conclusions in Section 5.

2 Related Work

As we mentioned before, a considerable part of the research effort on nested NER has been focused on English documents in the biomedical domain, mainly due to the availability of corpora such as GENIA [1], which contains MEDLINE abstracts annotated with nested NE's. Aiming to reuse well known, successful techniques, common approaches to nested NER treat the problem either as a separate postprocessing stage of the classic NER problem or as the combination of several instances of the classic problem.

The first type of approaches consists in extending classic NE recognizers through a mechanism for merging together several NE's or detecting new NE's that either are embedded in the original ones or contain them. For instance, Zhang et al. [2] address English biomedical NER by using Hidden Markov Models (HMM's) to recognize the innermost NE's. As a postprocessing stage, they apply a set of rules extracted from the training data to detect other NE's that contain the initial ones.

The second type of approaches consists in handling different nesting levels or NE types as separate problems by combining several instances of a classic recognizer. Zhang et al. [2] also propose a method consisting in the application of several passes of an HMM-based recognizer, adding a new nesting level in each one. The process starts by recognizing the innermost phrases. After a pass is completed, the input sequence is modified in such a way that recognized entities are treated as special tokens, and then a new pass is performed to obtain a new nesting level. The process ends when no new NE's are found after a pass is completed. As the authors point out, errors in each pass affect the recognition process in further passes. Alex et al. [3] discuss three modeling techniques using the BIO encoding in English biomedical NER. Two of these techniques fall into this type of approach: *layering*, where each nesting level is modeled as a separate

BIO problem; and *cascading*, where entity types are divided into groups and a separate model is trained for each group. Layering is affected by the same situation mentioned before: errors in a level negatively impact the recognition process for further levels; whereas cascading is unable to recognize NE's that are embedded in other NE's of the same type.

The third modeling technique discussed by Alex et al., *joined label tagging*, does not belong to any of these two types of approaches. It consists in creating a new tagging scheme, where BIO tags for all nesting levels are concatenated to process all levels in a single pass. As they point out, this technique is prone to be affected by data sparseness. A similarity between this idea and our approach is that the recognition process for all nesting levels is performed in a single pass. However, Alex et al. address the problem by tagging a word sequence using a modification of the BIO tagset, whereas we classify entire phrases instead of independent words. Besides, they follow the order of words in the sentence, whereas we apply a syntax-dependent tree traversal-based strategy.

For Spanish, due in part to the lack of annotated corpora containing nested NE's, a small number of works have addressed the problem. The MICE system [4] relies on the distinction between strong NE's, which contain proper nouns, and weak NE's, which are constructions containing trigger words and optionally other embedded NE's. AdaBoost is used to recognize and classify strong NE's whereas a handcrafted context independent grammar is used to recognize weak NE's. A complete quantitative evaluation of this method is not presented in [4] due to the early stage of the corpus development.

Recently, within the framework of SemEval 2007 Task 9 [5], Spanish and Catalan nested NER was addressed as a subtask. The UPC system [6] was the only one that submitted results for NER. Based on the distinction between strong and weak NE's that characterizes the competition corpus, UPC performs NER in two stages, both using AdaBoost. The first stage deals with the classification of strong NE's, whereas the second deals with the detection and classification of weak NE's. UPC, as well as MICE, are examples of the first type of approaches to nested NER mentioned earlier.

The two types of approaches to nested NER that we have discussed here tackle the complexity of the problem by splitting it into subproblems that are solved separately. However, in doing this, it is possible that useful interactions between these subproblems are lost. We consider that approaches intending to find near-to-globally optimal solutions may exploit these interactions in order to obtain better results.

3 Our Approach

In the proposed method, the recognition process is carried out in a sentence by sentence basis. For each sentence, candidate phrases, i.e., those that may be NE's, are detected in the deep constituency tree. Candidate phrase detection is performed based on the syntactic labels of phrases, following the criterion that any definite noun phrase may be considered as a candidate. The set of candidate

phrases is represented using a tree, where each node contains the representation of one phrase and nesting is encoded by the parent-child relation between nodes. A set of boolean features is used to describe phrases, thus the representation of a phrase is a boolean vector containing the results of evaluating these features on a vicinity of the phrase in the deep constituency tree. Using a strategy based on the postorder traversal of the representation tree, the classification of the set of candidate phrases is carried out in such a way that the set of classes given to all candidate phrases in a nested structure approximates the global optimum. Next we describe this process in detail.

3.1 Obtaining the Representation

Let $S = w_1 w_2 ... w_n$ be a sentence. Consider two phrases $P = w_i w_{i+1} ... w_{i+k}$ $(i \geq 1,\ i + k \leq n)$ and $P^{'} = w_j w_{j+1} ... w_{j+k'}$ $(j > 1,\ j + k^{'} < n)$ contained in S. If $j > i$ and $j + k^{'} < i + k$, we say that $P^{'}$ is embedded in P. If no phrase $P^{''}$ embedded in P exists such that $P^{'}$ is embedded in $P^{''}$, we say that $P^{'}$ is immediately embedded in P.

For each sentence, once the deep constituency tree has been obtained and candidate phrases are detected in it, the representation tree is constructed in such a way that each candidate phrase is represented by a node. If the node N represents phrase P, its children are those nodes $N_1, N_2, ..., N_e$ representing phrases $P_1^{'}, P_2^{'}, ..., P_e^{'}$ that are immediately embedded in P. Unless the whole sentence is a candidate phrase, an artificial root node is added such that all nodes representing candidate phrases that are not immediately embedded in any other are its children.

For example, consider the sentence *Agüero visitó Trinidad y Tobago* (*Agüero visited Trinidad and Tobago*). The named entities contained in that sentence are *Agüero* (person name), *Trinidad y Tobago* (location name), *Trinidad* and *Tobago* (location names embedded in *Trinidad y Tobago*). Figure 1 shows the representation tree for that sentence. For clarity, in the figure every node (except the artificial root node) is labeled as *Repr: P*, where P is the candidate phrase represented by it. Notice that what the nodes in the tree actually contain are boolean vectors representing the phrases.

Let $\wp$ be the set of all possible candidate phrases. A phrase P will be represented in terms of a set of q boolean features $f_i : \wp \rightarrow \{0, 1\}$ $(1 \leq i \leq q)$. Thus, the representation of phrase P is a boolean vector $r = (f_1(P), f_2(P), ..., f_q(P))$.

We consider two types of features. The first one includes features that are evaluated locally, i.e., on the phrase per se. We use POS tags, trigger word dictionaries and gazetteers of location and organization names.

The second type are syntax-dependent features. A set of these features checks whether the syntactic function of the candidate phrase or, alternatively, that of a prepositional phrase in which it is immediately embedded, is one of the following: subject, direct object, indirect object, adjunct or agent complement. Additionally, a set of verb lemma dictionaries is collected from the training corpus and a feature is defined for each dictionary. Each one of these features checks whether the lemma of the verb governing the clause where the candidate phrase

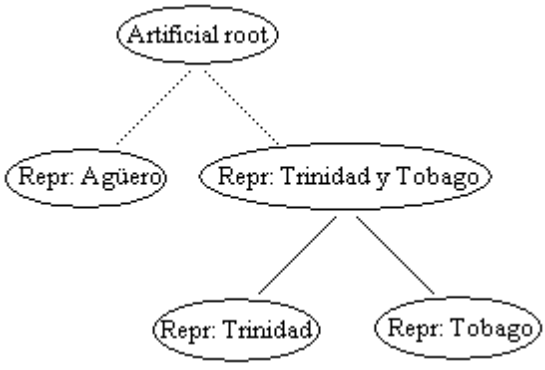


Fig. 1. Representation tree obtained for the sentence *Agüero visitó Trinidad y Tobago*

is embedded occurs in its associated dictionary. In order to collect the set of dictionaries, a mapping is constructed between entity types and syntactic functions. Thus, dictionaries are constructed for "person as subject", "organization as object", and so on. For each entity type T and each syntactic function F, a dictionary is constructed which contains the lemmas of verbs occurring in at least one clause in the training corpus where a NE of type T is found being its syntactic function, or that of a prepositional phrase where it is immediately embedded, F. Auxiliary verbs are disregarded.

For example, when obtaining the representation for the phrase *Agüero* in the sentence considered in Figure 1, suppose a feature f_i is associated with a dictionary containing the verb *visitar* (lemma of *visitó*). This feature will yield the value 1 when evaluated on the phrase because it belongs to a clause governed by *visitó*. In constructing the dictionaries, if this sentence were a part of the training data, the dictionary corresponding to the mapping "person as subject" would contain *visitar*, because *visitó* is the verb governing the clause and the syntactic function of *Agüero* (a person name) is subject.

The combination of syntactic function information and verb lemma dictionaries is expected to help take into account the behavior of candidate phrases. Intuitively, verb lemma dictionaries are expected to provide information about the action in which the phrases are involved and the syntactic function is expected to provide a rough approximation of the role that each phrase is playing.

While locally evaluated features are expected to help determine the literal type of a NE, syntax-dependent features should allow to discriminate between literal and metonymic readings of NE's.

3.2 Traversing the Representation Tree

In selecting an appropriate order to traverse the representation tree, we consider the following ideas. First, in determining the type of a NE, it is convenient that the types of its embedded phrases are considered. Second, since errors in

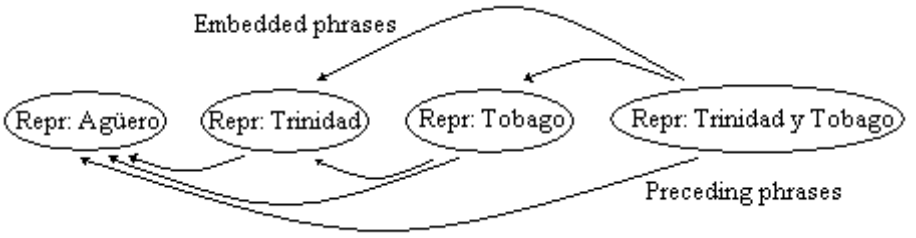


Fig. 2. Dependences expected to be captured by the postorder traversal strategy

irrevocably classifying a phrase cause a negative impact in the classification of the phrases in which it is embedded, the best classification is the one that satisfies some global optimality measure.

In order to allow the syntactic structure of the sentence to guide the classification process, we propose a strategy that consists in generating a sequence of candidate phrase representations following a postorder traversal of the representation tree. The sequence thus obtained is such that the node representing a phrase is located after those representing its embedded phrases and after those representing preceding phrases. Figure 2 illustrates the situation for the previously analyzed sentence.

3.3 Classification

We treat the classification of the set of candidate phrases as a sequence classification problem by using a Markovian approach, according to which only one independence assumption needs to be inserted. A Viterbi search is performed on that sequence to capture the desired dependences of a candidate phrase to its embedded phrases, as well as those to previoulsy occurring phrases, which are commonly used in Markovian approaches. We combine all candidate phrases in a single sequence and approximate a globally optimal solution without making the classification of any level irrevocable. This aims to avoid the problems of iteratively classifying superimposed nesting levels pointed out in Section 2.

For the classification process, the class set C contains one class for each NE type, plus an extra class *NONE* to handle candidate phrases that are not NE's.

Given a sequence of boolean vectors $r_1, r_2, ..., r_m$, which represent candidate phrases obtained following the postorder traversal of the representation tree, we use a Conditional Markov Model (CMM) to obtain the class sequence $c_1, c_2, ..., c_m$ that maximizes the probability $P(c_1, c_2, ..., c_m | r_1, r_2, ..., r_m)$. We follow the independence assumption adopted by Punyakanok and Roth [7]:

$$P(c_t | r_t, ..., r_1, c_{t-1}, ..., c_1) = P(c_t | r_t, c_{t-1}) \quad (1)$$

That is, at step t, the probability of the phrase represented by r_t to be classified as c_t depends on the representation itself and on the class given to the phrase represented by the previous vector in the sequence.

In their work, Punyakanok and Roth split $P(c_t|r_t, c_{t-1})$ into $|C|$ functions $P_{c_{t-1}}(c_t|r_t)$. Here we do not follow this approach to prevent class unbalance from causing data sparseness in estimating $P_{c_{t-1}}(c_t|r_t)$ for some values of c_{t-1}. Instead, we codify c_{t-1} in terms of $|C|$ additional features $f'_{q+j} : C \rightarrow \{0, 1\}$ $(1 \leq j \leq |C|)$ such that

$$f'_{q+j}(c_{t-1}) = \begin{cases} 1 \text{ if } c_{t-1} = c_j \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

and obtain a new vector $r'_t = (f_1(P_{r_t}), ..., f_q(P_{r_t}), f'_{q+1}(c_{t-1}), ..., f'_{q+|C|}(c_{t-1}))$, where P_{r_t} is the phrase whose representation is r_t. The new vector is an extension of r_t, which allows us to express $P(c_t|r_t, c_{t-1}) = P(c_t|r'_t)$.

We use the variant of the Viterbi algorithm described by Punyakanok and Roth to approximate the globally optimal solution.

In this paper, we consider two variants of our postorder traversal-based classification strategy. The first one consists in constructing a single sequence containing all the nodes in the representation tree, except the root node when it was artificially added. In what follows, we will refer to this variant as sentence-level postorder traversal-based strategy. The second variant limits interactions to those between phrases that belong to the same nested structure. In this variant, for the most common case when the root node is artificial, a separate sequence is obtained for each subtree corresponding to a child node of the root. Each sequence thus obtained is processed independently during both training and classification. Even though the whole sentence is not processed at once by using this variant, contextual information may still be captured if syntax-dependent features are used in describing phrases. We will refer to this variant as nested structure-level postorder traversal-based strategy. Notice that if the root is not an artificial node, both variants are equivalent.

Probability estimation is carried out using Maximum Entropy (ME). For describing phrases, ME relies on a set of functions that depend not only on the result of evaluating a boolean feature on the phrase but also on the class for which the probability is estimated. We integrate our feature set into the ME framework following the idea described by McCallum et al. [8]. Thus, for each component $r'_{t,i}$ $(1 \leq i \leq q + |C|)$ of the extended representation vector r'_t and each class $c_j \in C$, a function

$$g_{i,j}(r'_t, c_t) = \begin{cases} 1 \text{ if } r'_{t,i} = 1 \text{ and } c_t = c_j \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

is constructed. We train the estimators using Generalized Iterative Scaling [9].

4 Experiments

The purpose of our experiments is three-fold. Firstly, we evaluate the usefulness of syntax-dependent features for describing candidate phrases. Secondly, we compare the behavior of the postorder traversal-based classification strategy, both

the sentence-level and the nested structure-level variants, against two bottom-up strategies using the same probability estimator. Finally, we compare our results to those reported for the NER subtask at SemEval 2007 Task 9 [5]. The Spanish dataset for that subtask is used throughout our experiments.

4.1 Experimental Setting

The Spanish dataset used for SemEval 2007 Task 9 is a subset of the CESS-ECE corpus [10]. This subset contains 101,136 words in 3,611 sentences. The corpus is annotated with POS tags, lemmas, syntactic constituents, syntactic functions, named entities, verb argument structure, thematic roles, semantic classes of verbs and WordNet synsets for the 150 most frequent nouns. The training/test corpus size ratio is 9:1. The test corpus is furtherly split into two subsets, in-domain and out-of-domain test corpora. The in-domain test corpus contains documents covering the same domain as those of the training corpus, whereas the out-of-domain test corpus contains documents from a different domain in order to assess the systems adaptability.

Six types of named entities are annotated, namely person, organization, location, number, date and other. Table 1 shows the distribution of the different entity types throughout the corpus. In the table, *test.in* stands for in-domain test corpus whereas *test.out* stands for out-of-domain test corpus.

4.2 Results and Discussion

Evaluation was carried out in terms of precision, recall and F_1 measures in the same way that it was done in SemEval 2007 Task 9. Table 2 shows the results obtained in our experiments.

The first section of the table contains the results for several variations of the tree traversal-based strategy. **PO_sent** represents the sentence-level postorder traversal-based strategy, whereas **PO_struct** represents the nested structure-level postorder traversal-based strategy.

BU_glob stands for a bottom-up strategy where two indepedence assumptions are considered. First, the class of a node is assumed to depend only on the set of classes of its children. All class combinations are considered. Since the average number of children of non-leaf nodes in the representation trees obtained

Table 1. Distribution of NE types in the Spanish corpus for SemEval 2007 Task 9

NE type	training	test.in	test.out
person	1,953	116	72
organization	1,346	234	6
location	944	173	67
number	758	95	5
date	500	58	6
other	769	95	26

Table 2. Experimental results over the Spanish dataset of SemEval 2007 Task 9 corpus

	In-domain			Out-of-domain			Global		
	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
PO_struct_LS	74.48	70.04	72.19	68.79	53.30	60.06	73.56	66.84	70.04
PO_sent_LS	73.76	69.26	71.44	67.61	52.75	59.26	72.75	66.11	69.27
BU_glob_LS	70.01	68.74	69.37	63.76	52.20	57.40	68.98	65.58	67.24
BU_loc_LS	67.60	69.00	68.26	61.49	54.40	57.73	66.56	66.21	66.39
PO_sent_L	64.94	63.68	66.67	64.86	52.75	58.18	69.06	61.59	65.11
PO_struct_L	69.89	63.81	66.71	64.19	52.20	57.58	68.90	61.59	65.04
BU_loc_L	65.75	68.48	67.09	56.04	56.04	56.04	63.96	66.11	65.02
BU_glob_L	68.65	64.20	66.35	61.49	54.40	57.73	67.35	62.33	64.74
UPC	72.53	68.48	70.45	62.03	53.85	57.65	70.65	65.69	68.08
baseline	-	-	-	-	-	-	71.88	12.07	20.66

from the test corpus is 1.34 and the maximum is six, running times for this method still remain reasonable in spite of its asymptotically exponential nature. To prevent data sparseness in considering a set of classes, the second assumption consists in considering the classes of all children of a node mutually independent. Under these considerations, a bottom-up dynamic programming method is used over each nested structure to approximate a globally optimal set of classes for all nodes.

On the other hand, **BU_loc** stands for a bottom-up strategy where no dynamic programming is used. In classifying a node, the locally most probable class according to the estimator is taken. The class given to a node is considered when classifying its parent but the classification process is iterative and irrevocable. This may be seen as an instance of the layering modeling technique [3] discussed in Section 2.

The suffix **L** means that only locally evaluated features (POS tags, trigger word dictionaries and gazetteers) were used to describe the candidate phrases. The suffix **LS** means that both locally evaluated and syntax-dependent features (syntactic functions and verb lemma dictionaries) were used.

The second section of the table contains results reported for the NER subtask at SemEval 2007 Task 9. UPC stands for the only system presented to the competition [6]. Additionally, the baseline results for the competition are presented. The baseline consisted on collecting a gazetteer of NE's from the training corpus and recognizing those segments in the test corpus that had the longest match with some of the NE's in this gazetteer.

Columns 2 to 4 show precision, recall and F_1, in that order, for the in-domain test corpus. Similarly, columns 5 to 7 show the results for the out-of-domain test corpus and columns 8 to 10 for the entire test corpus. In each section, results are sorted by global F_1. Only results on the entire corpus are available for the baseline.

Several observations can be made after analyzing these results. First, we can corroborate the possitive impact of using syntax-dependent features to describe candidate phrases for the classification process. Global F_1 values using both

locally evaluated and syntax-dependent features are always greater (between 1.37% and 5%) than those where only locally evaluated features were used.

A second important observation is that the postorder traversal-based classification strategy outperformed the bottom-up strategy in all cases sharing the same feature set. We consider that the main reason for this phenomenon is that the latter includes more independence assumptions than the former. Notice that when syntax-dependent features are used, the differences between global F_1 values are greater (between 2.03% and 3.65%) than those for the case when only locally evaluated features are used (between 0.02% and 0.31%). This suggests that syntax-dependent features enhance the postorder traversal-based classification strategy.

Regarding the postorder traversal-based strategy, we can also observe that when syntax-dependent features are used, the nested structure-level variant works better than the sentence-level variant whereas the opposite situation occurs when only locally evaluated features are used. Although we consider that these results cannot be taken to be conclusive in that aspect as yet because the differences are not very considerable, there are some ideas that might explain the situation. When only locally evaluated features are used, constructing a single sequence for the whole tree may help the classification process by providing more contextual information. On the other hand, syntax-dependent features contain contextual information per se and, in this case, it is possible that some amount of noise is added if we also consider dependences between phrases that may not be directly related by the clause-governing verb.

As can be largely expected, results over the in-domain corpus are better than those over the out-of-domain corpus in all cases.

Finally, when syntax-dependent features were used, the two variants of the postorder traversal-based classification strategy outperformed the results reported for the UPC system that participated in the SemEval competition. Regarding the competition baseline, the conservative nature of the method results in very low recall, which causes the F_1 value to be very low. According to F_1 values, this baseline is largely outperformed by the proposed method as well as by the UPC system. However, it should be noticed that UPC obtains a precision value below that of the baseline, whereas our syntax-dependent postorder traversal-based method outperforms the baseline in both precision and recall.

5 Conclusions

In this work, Spanish nested NER has been addressed. Unlike most approaches to nested NER, our method treats the classification of all phrases in a nested structure as a single problem in order to obtain a near-to-globally optimal solution. We propose a tree representation for the set of candidate phrases in which syntactic information, both structural and functional, is encoded. In our opinion, the main contribution of this work is the proposal of a global classification strategy based on the postorder traversal of this representation tree.

Experimental results on the Spanish dataset for the SemEval 2007 Task 9 NER subtask show the validity of our postorder traversal-based classification

strategy and the usefulness of syntactic information in describing phrases for the classification process.

In order to improve these results, an attractive direction for future work is the use of semantic information, such as verb senses, verb semantic classes, noun senses for trigger words, semantic roles, etc., in order to turn syntax-dependent features into semantic features. For example, if verb sense information is combined with verb lemma dictionaries in an appropriate way, results are likely to improve. Similarly, it may be helpful to use semantic roles instead of (or in combination with) syntactic functions.

References

1. Ohta, T., Tateisi, Y., Kim, L., Mima, H., Tsujii, J.: The GENIA corpus: An annotated research abstract corpus in molecular biology domain. In: 2nd International Conference on Human Language Technology Research, San Diego, CA, USA, pp. 82–86 (2002)
2. Zhang, J., Shen, D., Zhou, G., Su, J., Tan, C.: Enhancing HMM-based biomedical named entity recognition by studying special phenomena. Journal of Biomedical Informatics 37, 411–422 (2004)
3. Alex, B., Haddow, B., Grover, C.: Recognising Nested Named Entities in Biomedical Text. In: BioNLP Workshop on Biological, translational, and clinical language processing, Prague, Czech Republic, pp. 65–72 (2007)
4. Arévalo, M., Civit, M., Martí, M.A.: MICE: A module for Named Entities Recognition and Classification. International Journal of Corpus Linguistics 9(1), 53–68 (2004)
5. Márquez, L., Martí, M.A., Taulé, M.: SemEval-2007 Task 09: Multilevel Semantic Annotation of Catalan and Spanish. In: 4th International Workshop on Semantic Evaluations, Prague, Czech Republic, pp. 42–47 (2007)
6. Márquez, L., Padró, L., Surdeanu, M., Villarejo, L.: UPC: Experiments with Joint Learning within SemEval Task 9. In: 4th International Workshop on Semantic Evaluations, Prague, Czech Republic, pp. 426–429 (2007)
7. Punyakanok, V., Roth, D.: The use of Classifiers in Sequential Inference. Advances in Neural Information Processing Systems 13, 995–1001 (2001)
8. McCallum, A., Freitag, D., Pereira, F.: Maximum Entropy Markov Models for Information Extraction and Segmentation. In: 17th International Conference on Machine Learning, Stanford, CA, USA, pp. 591–598 (2000)
9. Darroch, J., Ratcliff, D.: Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics 43, 1470–1480 (1972)
10. Martí, M.A., Taulé, M., Màrquez, L., Bertran, M.: CESS-ECE: A Multilingual and Multilevel Annotated Corpus (2007), http://www.lsi.upc.edu/~mbertran/cess-ece/publications

Question/Answering Clarification Dialogues

Luis Quintano and Irene Pimenta Rodrigues

Departamento de Informática
Universidade de Évora, Évora, Portugal
ljcq@uevora.pt, ipr@di.uevora.pt

Abstract. The presented (Question Answering) system uses clarification dialogues in order to disambiguate the possible meaning of user questions in natural language. The system will only query the user when commitment to some meaning is relevant for the final answer. When there are multiple different answers to a user query, the system initiates a clarification dialogue trying to ascertain the user's intentions. The clarification question is always about a relevant property of a (user) discourse entity. The answer to the clarification question (its semantic representation) can be incorporated into the initial user question enabling its reinterpretation under the new constraints. This is developed in an integrated logic programming framework, based on constraint logic programming using the GnuProlog(-cx) language [3,7] and the ISCO framework [2]. The use of this LP framework allows the integration of Prolog-like inference mechanisms with classes and inheritance, constraint solving algorithms and provides the connection with relational databases, such as PostgreSQL.

1 Introduction

We present a Question Answering system that provides a natural language interface to Databases. Our system uses clarification dialogues in order to disambiguate the possible meaning of the user questions when its necessary: when there is more then one answer to a user question.

The system (NL-UE) was built to interface the IIS-UE (Universidade de Évora Integrated Information System) that has relevant information for students (enrolled courses, grades, class summaries, etc.), for teachers (courses information, projects, students evaluation, personal data, etc.) and staff (data management, statistics, personal data, etc.).

There are dedicated applications built to manage the school information. To use these applications users must be aware of some structural details. A student may know what information he wants but he doesn't know how to get it from the existent applications. A natural language QA application is a major enhancement to overcome these difficulties.

The IIS-UE information is stored in Postgresql relational databases [6]. To access this data, NL-UE must map the user question to a database query language so that the resulting pragmatic interpretations can be evaluated.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 155–164, 2008.

NL-UE implementation is based on constraint logic programming using the GnuProlog(-cx) language [7] and the ISCO framework [2].

The system architecture is simple and module-based. The system has five distinct modules which are connected through well defined API's:

- Syntactic analysis. Each user question is represented by a set of syntactic trees.
- Semantic analysis. Each syntactic tree of the user question is transformed into a DRS (Discourse representation structure)[8].
- Semantic/pragmatic interpretation. Each DRS is interpreted in the database context. Discourse referents are transformed into constraint database entities and Discourse conditions are transformed into one or more database relations, giving rise to none, one or more DRS-pragmatic for each DRS-semantic.
- Evaluation Module. This module evaluates each DRS-pragmatic by building sets of constraint Discourse entities, at least one for each DRS-pragmatic representation.
- Dialogue manager - This module will look at the sets of constraint Discourse entities. If there is only one, it is the system answer. If there is more than one, it will use an heuristic to choose a question to the user. The user answer will be added to the DRS of the initial question. This DRS-semantic will be processed by the semantic pragmatic interpreter.[1]

In the interpretation of a user question, ambiguity may arise from the syntactic analysis, namely due to pp-attachment, or in the semantic-pragmatic analysis, namely due to proper nouns resolution or due to the existence of more than one way to relate two entities in the database.

For instance, a question such as the one bellow will have only one syntactic analysis, one DRS-Semantic representation but multiple DRS-Pragmatic ones due to the proper noun interpretation (Management and Economics):

Which Management teacher lectures Economics?

For instance, in our database both 'Management' and 'Economics' may refer to a Department, to a degree or to a course. However we will only get five DRS-pragmatic representations because it is not possible to relate a 'teacher' to a 'Department' with the relation 'lectures' in our Database.

If the set of teachers is different for each of the DRS-pragmatics the system will initiate a clarification dialogue. The only way to avoid this ambiguity would be using the full name of the entity the user intends to refer, but often the user does not know the full name to use and does not know that the name he used may be ambiguous.

Other QA system to interface databases such as Start [10], Precise [11], Masque/SQL [5], or Wow[9] do not have the ability of detecting when it is

[1] After reaching a solution the evaluation context is reset. Follow-up questions are not considered.

relevant to clarify the user intentions or of finding a pertinent question to clarify the user intentions.

These systems transform the user question in a representation capable of querying a database (Xquery, Xpath or SQL). The user answer will be the result of the evaluation of the query built by the systems. If more than one answer is found all of them are shown to the user as possible answers, or the queries are displayed for the user to choose.

When a user does not know the database structure and the associated Entity-Relation model it will be difficult to choose between the database queries.

For these systems it is difficult to add new constraints to the representation of a user query because they do not build a semantic representation of the user query. Since they translate from the sentence syntactic tree directed to SQL-like language they loose the capability of adding new constraints (conditions) to the discourse entities.

In this paper we present clarification dialogues as a tool for user questions disambiguation in a QA system, focusing on the relevancy of the system answers [2]. Our main goal is to correctly interpret the user intentions and build a coherent and relevant answer or, in other terms, an answer the user would expect. The process of determining the correct interpretation can be straightforward or may lead to ambiguity problems which need to be solved.

We claim that the quality of the dialogue, or more specifically the quality of the answers built by the system, depend on the quality of the disambiguation process.

Section 2 describes the followed approach in a locally developed database QA system and how it benefits from the development framework that was used. In section 3 we motivate and present the clarification process. Then (section 4) some practical results and conclusions are shown.

2 NL-UE

Work developed in Universidade de Évora resulted in a portuguese dialogue interface prototype[3] for the institution information repository [12] [4].

Development was made using a logic-based framework (ISCO) [2] which enables a more declarative pragmatic representation (in opposition to a direct SQL representation, Xpath or Xquery) with direct access to relational databases.

The system architecture contains five distinct modules connected through well defined API's [fig. 1].

The syntactic parser is chunk-based [1] imposing no strong restrictions to the query sentence structure and leaving all possible interpretation analysis and filtering to the semantic/pragmatic parser [4]. Semantic interpretation is built using First-Order Predicate Logic.

[2] A system answer may be a clarifying question (in the user perspective) or a final answer.

[3] Although examples are shown in English for better understanding.

[4] The number of possible interpretations is directly influenced by the number of attached PP's.

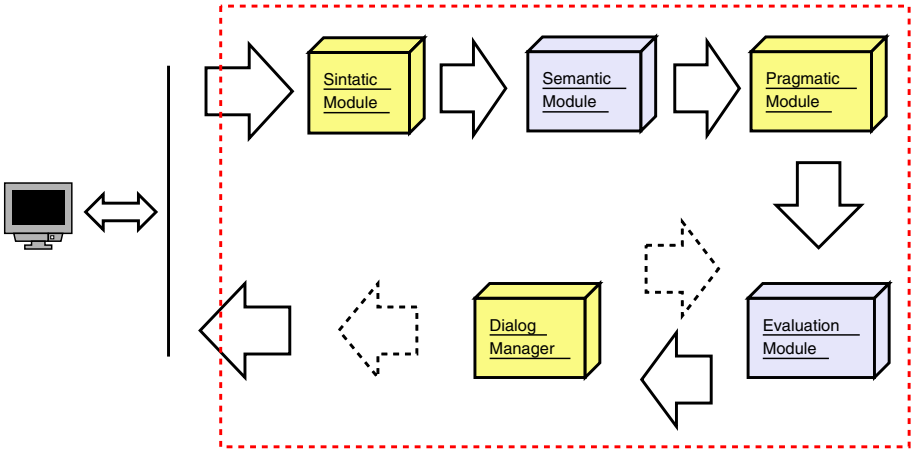


Fig. 1. System Architecture

Semantics:

```
teacher(A),rel(A,B),name(B,'Management'),lectures(A,C),name(C,'Economics')
```

with three distinct discourse referents:

```
A: (_-singular)  B: (female-singular)  C: (female-singular)
```

Fig. 2. Semantic interpretation results

Ex1: *Which Management teacher lectures Economics?*

This question has one semantic [fig. 2][5] and three pragmatic interpretations [fig. 3], each one of them corresponding to a distinct "view" of the user question:

(1) *The teachers that lecture a course which contains (in its name) "Management" and "Economics"*

(2) *The teachers that belong to a "Management entity" (organizational unit) and lecture an "Economics" curriculum*

(3) *The teachers that belong to a "Management entity" (organizational unit) and lecture an "Economics" course*

Transition to pragmatic interpretation is done by adding contextual information with the instantiation of referents. The final representation is in the ISCO language, enabling a direct evaluation in the relational database.

Evaluation will define a set of possible answers. If no ambiguity is found the dialogue manager will directly answer the user with the only possible answer. If more than one possible answer is found, then the dialogue manager enters in a clarification process [Fig. 4][6].

[5] The example refers to the Portuguese interpretation.

[6] Restraining solutions is possible by using GnuProlog FD Domain Solver.

Pragmatics:

```
(1) teacher(A,_), course(B), lectures(A, B).

A in [26885..47787] - male/female - singular
B in [188:1270:1320:1688:1689:2479] - female - singular
C = B

(2) teacher(A), organizational_unit(B), belongs_to(A,B), curriculum(C),
    lectures(A, C).

A in [26885..47787] - male/female - singular
B in [50981:51005:51015:51019..51020:51023:51034:51040:51050:51104:
      51135:51186..51187:51197:51200:51231] - female - singular
C in [51250:51293] - female - singular

(3) teacher(A), organizational_unit(B), belongs_to(A, B), course(C),
    lectures(A, C)

A in [26885..47787] - male/female - singular
B in [50981:51005:51015:51019..51020:51023:51034:51040:51050:51104:
      51135:51186..51187:51197:51200:51231] - female - singular
C in [187..188:452:646:797:847:1132:1160:1174:1206:1270:1320:1353:
      1496:1544:1688..1690:1787:1834:1857..1858:1860..1863:1865..1867:
      1919:2195..2203:2478..2479:2488:2491:2493..2494:2657:2672..2673:
      2687:2783:2819:2821:2874:2892:2895:2998:3029..3030:3113:3118..
      3119:3223:3375:3630:3646:3656:3682:3755] - female - singular
```

Fig. 3. Pragmatic interpretation results

```
while not a terminal condition:
  collects properties for each referent
  proceeds with heuristic evaluation
  chooses the best property (best heuristic punctuation)
  questions the user / receives the answer
  restrains solutions to the question result
end while
```

Fig. 4. Dialogue Manager: clarification algorithm

A `terminal` condition will be reached if only one answer is achieved as is illustrated by the following dialogue example.

```
USER: "Which Management teacher lectures Economics?"

SYSTEM: "Management" refers to a:
1) Course
2) Organizational Unit
USER: 2
```

```
SYSTEM: "Economics" refers to a:
1) Course
2) Curriculum
USER: 2

teacher: Ana Silva Nunes,Luis Calado Pereira,Maria do Rosário
Fernandes,... organizational_unit: Management department
curriculum: Economics
```

3 The Need for Clarification

Distinct semantic interpretations (DRS-Semantic) may be:

- **eliminated**, because they have no meaning in the application context [7]
- **merged**, because distinct interpretations may lead to the same pragmatic meaning

The final interpretation set may lead to ambiguity problems. These problems are only relevant if (after evaluation) more than one possible answer is found. Ambiguity may appear in distinct forms:

- **association ambiguity**, if possible interpretations associate differently the sentence referents.
- **concept ambiguity**, if possible interpretations refer distinct types of entities for the same sentence referent.
- **entity ambiguity**, if possible interpretations refer distinct entities for the same sentence referent.

Each sentence has distinct ambiguity problems which have to be solved. The most common is **entity ambiguity**. Very often proper nouns are only partially used in sentences. It's not frequent to refer to an entity by its full description (name).

Ex2: *Who teaches Databases?*

In this example **entity ambiguity** is reflected in *databases* (there may be several *databases* courses).

```
USER: "Who teaches Databases?"

SYSTEM: "Databases" is:
1) Databases
2) Advanced Topics in Databases
USER: 1
```

[7] Application context is defined by the information repositories structure and data. The used framework enables direct access to relational structure and data for contextual validation.

```
SYSTEM:
teacher: Teresa Alves,Irene Ramalho
course:  Databases
```

Ex3: *Which teacher lectures Mathematics?*

This question presents us a distinct kind of ambiguity: **concept ambiguity**. *Mathematics* may refer to a course, to a curriculum or even to a department. This is a typical example where clarification dialogue is needed so that the system can understand the user intentions and find the correct interpretation.

```
USER: "Which teacher lectures Mathematics?"

SYSTEM: "Mathematics" is a:
1) curriculum
2) course
USER: 2

SYSTEM: "Mathematics" belongs to the:
1) Management department
2) Educational department
3) Mathematics department
USER: 1

SYSTEM:
teacher: Rui Manuel Dias
course: Financial Mathematics
```

Ex4: *Who teaches the curriculum of Chemistry?* [8]

Besides the ambiguity problems raised by the *databases* concept, another ambiguity type is found: **association ambiguity**:

1. *Who teaches the curriculum of Chemistry?*, meaning that the user intends to know who are the teachers of *curriculums* that are somehow related to a *Chemistry* entity.
2. *Who teaches the curriculum of Chemistry?*, meaning that the user intends to know who are the teachers of *Chemistry* (which is a curriculum).

Even if they make no sense, all associations of the PP "of Chemistry" are considered, as long as it is possible to map them as valid database queries. In this particular example the first interpretation, although making no common sense, creates valid queries requiring further clarification:

```
USER: "Who teaches the curriculum of Chemistry?"

SYSTEM: "Chemistry" is a:
```

[8] Although not correct, this English representation reflects in a better way the original Portuguese ambiguity problem.

```
1) curriculum
2) course
USER: 1

SYSTEM: "Chemistry" is:

1) Chemistry Engineering
2) Chemistry and Physics
3) Chemistry
USER: 1

teacher: Telma Silva,Ana Mendonça Marques,...
curriculum:  Chemistry Engineering
```

4 Practical Results and Conclusions

As no global evaluation of the system was made yet, the results shown are based on a collection of exemplifying questions over the Universidade de Évora information corpus [9].

	Sentence	Ambiguity
A	Which teacher lectures Financial Management?	`none`
B	Who teaches the Database course?	`entity`
C	Which teacher lectures Mathematics?	`conceptual + entity`
D	Does Pedro lectures the Physics course?	`entity + association`

While sentence **A** has no ambiguity problems, sentence **B** reflects `entity` ambiguity caused by the fact that exists more than one "database" entity.

Both sentence **C** and **D** reflect more than one ambiguity type. Besides the `entity` type, the first sentence (**C**) contains a reference to a "mathematics" entity which can be a course, a curriculum or a department (`conceptual` ambiguity). The second sentence (**D**) has `association` ambiguity in consequence of the possible connections of its PP attachment.

	A	**B**	**C**	**D**
Number of Pragmatic Interpretations	1	1	2	2
Average Execution Time (1)	1,408 s	1,276 s	1,526 s	1,503 s
Average Execution Time (2)	–	0,011 s	0,053 s	0,049 s
Average Number of Steps	0	1	1,8	2,4

`Average Execution Time (1)` refers to the time the system take to place the first question or to give an answer in case no clarification is needed. `Average Execution Time (2)` refers to the time the system take to re-evaluate a question

[9] More specifically on a sub-corpus of "teaching distribution" information (relates teachers with courses and curriculums).

based on a user clarification answer. `Average Number of Steps` refers to the number of clarification questions needed to reach the final answer. As no real users were used, answers were chosen randomly. The system was tested on a bi-processor Intel(R) Xeon(TM) CPU 3.40GHz with 2Gb of RAM memory.

This paper presents a Question Answering system that provides a natural language interface to relational databases. It uses clarification dialogue as a tool to disambiguate between distinct interpretations of a user question.

The clarification mechanism leads to a straightforward dialogue with the user. The chosen questions are the most relevant according to the actual interaction context and always intend to get to a solution with the least number of steps as possible. Heuristic evaluation is used to ensure the quality of the questions.

Development is made using a logic-based framework with a high-level API for querying generic databases, an important feature to ensure the system portability. The system is being developed under the IIS-UE context although recently it has been ported (with success) to another context (portuguese historical data).

NL-UE was designed to be easily ported to distinct contextual domains. Practical experience proved it. To work correctly, some features must be ensured within the database structure (entities should have a unique identifier, auxiliary relations must exist, etc). All this features are automatically implemented by NL-UE which makes the system almost 100% independent of the application domain.

The next steps will be to confirm the correctness of the followed methodology and to consolidate the system implementation. A user group evaluation is also essential to prove the system usefulness for the academic community.

References

1. Abney, S.P.: Parsing by chunks. In: Berwick, R.C., Abney, S.P., Tenny, C. (eds.) Principle-Based Parsing: Computation and Psycholinguistics, pp. 257–278. Kluwer, Dordrecht (1991)
2. Abreu, S.: Isco: A practical language for heterogeneous information system construction. In: Bartenstein, O., Geske, U., Hannebauer, M., Yoshie, O. (eds.) INAP 2001. LNCS (LNAI), vol. 2543. Springer, Heidelberg (2003)
3. Abreu, S., Diaz, D.: Objective: In minimum context. In: Palamidessi, C. (ed.) ICLP 2003. LNCS, vol. 2916, pp. 128–147. Springer, Heidelberg (2003)
4. Abreu, S., Quaresma, P., Quintano, L., Rodrigues, I.: A dialogue manager for accessing databases. In: Jaakkola, H., Kangassalo, H., Kawaguchi, E. (eds.) Information Modelling and Knowledge Bases XV. Frontiers in Artificial Intelligence and Applications, vol. 105, pp. 210–219. IOS Press, Amsterdam (June 2004)
5. Androutsopoulos, I., Ritchie, G., Thanisch, P.: Masque/sql: An efficient and portable natural language query interface for relational databases. In: Proc. of the Sixth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE 1993, Edinburgh, Scotland, pp. 327–330. Gordon and Breach (1993)
6. Development and T. User. The postgresql development team. postgresql user's guide (1996)
7. Diaz, D. (1999), `http://www.gnu.org/software/prolog`

8. Kamp, H., Reyle, U.: From Discourse to Logic. Kluwer, Dordrecht (1993)
9. Kardkovács, Z.T.: On the transformation of sentences with genitive relations to SQL queries. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 10–20. Springer, Heidelberg (2005)
10. Katz, B., Lin, J.J.: Start and beyond (2002)
11. Kautz, H., Etzioni, O., Popescu, A.: Towards a theory of natural language interfaces to databases. Intelligent User Interfaces (IUI) (2003)
12. Quintano, L., Rodrigues, I.P.: Using a logic programming framework to control database query dialogues in natural language. In: Etalle, S., Truszczyński, M. (eds.) ICLP 2006. LNCS, vol. 4079, pp. 406–420. Springer, Heidelberg (2006)

Using Graphs for Shallow Question Answering on Legal Documents*

Alfredo Monroy[1], Hiram Calvo[1], and Alexander Gelbukh[1,2]

[1] Center for Computing Research, National Polytechnic Institute
Mexico City, 07738, Mexico
[2] SoNet RC, University of Central Europe in Skalica, Slovakia
hcalvo@cic.ipn.mx, alopezm301@ipn.mx, gelbukh@gelbukh.com

Abstract. This work describes a Shallow Question Answering System (QAS) restricted to legal documents. This system returns a set of relevant articles extracted from several regulation documents. The set of relevant articles allows inferring answers to questions posed in natural language. We take the approach of representing the set of all the articles as a graph; the question is split in two parts (called A and B), and each of them is added as part of the graph. Then several paths are constructed from part A of the question to part B, so that the shortest path contains the relevant articles to the question. We evaluate our method comparing the answers given by a traditional information retrieval system—vector space model adjusted for article retrieval, instead of document retrieval—and the answers to 21 questions given manually by the general lawyer of the National Polytechnic Institute, based on 26 different regulations (academy regulation, scholarships regulation, postgraduate studies regulation, etc.); with the answer of our system based on the same set of regulations. The results show that our system performs twice as better with regard to the traditional Information Retrieval model for Question Answering.

1 Introduction

Modern search engines allow retrieving a set of documents given a set of keywords. After this step, the user should make an additional effort to filter them and to find the specific information that he or she needs [1]. In recent years, there has been an extensive research of alternative methods which allow to access information in an easier way than the one provided by the current search engines. Part of this research has focused in Question Answering Systems (QAS). These systems are classified with regard to the collection of documents which they access. If the collection is about several subjects, the system is called an Open Domain QAS; whereas for a single subject, it is called Restricted Domain QAS. Restricted Domain QAS (RDQAS) have received more attention lately: Several RDQAS have been developed, including voice recognition [2], focused in medical areas [3,4,5], domestic areas [2], construction [6], and legal [7,8].

Particularly for the legal domain, the workshop *Question Answering for interrogating legal documents* took place in 2003, in the framework of the JURIX Forum (The

* Work done under partial support of Mexican Government: SNI, SIP-IPN, COFAA-IPN, and PIFI-IPN.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 165–173, 2008.

foundation for Legal Knowledge Based Systems). Several works showed that a common problem is that traditional Information Retrieval Methods are not adequate to find the relevant fragments which answer legal questions because they do not consider the logical relationships between articles. In addition, many questions require an answer which cannot be found explicitly in a single article, or fragments of them, but intrinsically in the relationship between articles [9,10]. Some works use logic inference mechanisms such as COGEX System [14] and the system by Quaresma *et al.* [7]. However, these systems need expensive resources such as ontologies, axioms, and are language dependent. To avoid such requirements, we propose using a graph for capturing the relationships between articles in regulations answering with a set of articles related with the question and between them, hence the denomination Shallow QA—see Section 2. For details of the rest of the System, see Section 3.

Unfortunately, RDQAS do not have a standard evaluation system as the Open Domain QAS[1] have, mainly because of the heterogeneous characteristics of these systems. Still there is a long way for exploration in this subject. We test our system with regard to a traditional vector space model information retrieval system to answer questions particularly for the Spanish language given a set of 26 regulation documents from the National Polytechnic Institute. Details of our evaluation and experiments are given in Sections 4 and 5.

2 Using Graphs for Question Answering

We propose the architecture of the QAS based on the common characteristics posed by regulation documents, as well as the kind of questions and answers expected by the user. Regulation texts have a defined structure, they are composed of chapters, and these, in turn, are subdivided in articles. There exists a relationship among articles of a single regulation text and also among articles of different document.

The questions we focus on consist mainly on asking if it is possible to perform certain action or not, for example:

> Question: *Is it possible to award a honorable mention to a bachelor if he chose to graduate using the qualification option?*

For that kind of questions, a final 'yes' or 'no' is expected. However, particularly for the legal domain, such a short answer is not desirable, because the answer should be supported by its corresponding articles. For this reason, we focus in providing the set of articles from which it is possible to infer the final answer. For the previously stated question, the general lawyer of the National Polytechnic Institute, answers[2]:

> Answer: As stated by the Professional Graduation Regulation from the National Polytechnic Institute, the following can be concluded: The option of graduating by qualification proceeds if the student's average is higher than 9.0 and all of his subjects were approved in an ordinary way (Chapter II of "On Graduating Options", *article* 13). The candidate can only aspire to the award of honorific mention, if, in addition to covering other requisites

[1] For example, the TREC (trec.nist.gov) and CLEF (www.clef-campaign.org) competitions.
[2] www.abogadogeneral.ipn.mx/faqdlc.html

disposed in this regulation, he presents professional exam (Chapter VII, "On the Professional exam", *article* 43). Graduating by qualification does not require presenting the professional exam, so that it cannot be included within the article 43 fraction II of the mentioned Regulation; when this option is chosen, the candidate cannot obtain honorific mention.

We can see from this answer that it consists of the articles which support it, as well as the reasoning to obtain it. In our case, the developed system was designed to return, as previously mentioned, only the set of articles which eventually contain those which answer the posed question. Because our system does not return a completely logically evaluated answer such as 'yes' or 'no', it is considered as part of the Shallow Question Answering Systems.

It is considered that graphs are structures capable of coding naturally the similarity relationships between textual units, and also, its utility in natural language processing tasks has been verified [11]. Because of this, our system uses as fundamental component the representation of a set of documents as a graph where each node corresponds to an article of the set of documents, and the values associated to the edges correspond to the degree of similarity between them. Using this representation we pretend to reflect the structure of the regulation texts, and in turn, use it for the extraction of the articles which constitute the answer to a question to the system.

The fundamental parts of the system are shown in the graph in Figure 1. Question pre-processing consists on constructing the query based on the question, and Answer Extraction consists on adding the generated query as two new nodes (A and B). Then the shortest path between A and B is sought. We will show that this path contains articles highly related to the question, and they share certain degree of similarity between them.

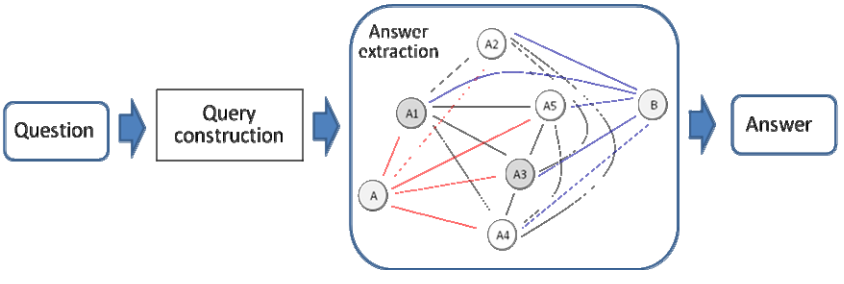


Fig. 1. General architecture of the proposed Question Answering System

3 System Description

3.1 Document Collection Preprocessing

The first step is a pre-processing of the set of documents which consists on articles (1698 articles in total) from 26 different regulations: First, convert to lowercase, eliminate punctuation symbols, roman numbering, Arabic numbers, hyphens, isolated chars from a to z; then, eliminate stopwords. This step was optional, and we conducted our tests with both keeping and removing stopwords.

3.2 Graph Description

The documents (articles of regulations) were represented as vectors, following the Vector Space Model [12, 13]. Each term was weighted by the TF·IDF measure (Term Frequency—Inverse Document Frequency). See equations (**1**), (**2**) and (**3**).

$$tfidf = tf_{t,j} \cdot idf_i \tag{1}$$

$$tf_{t,j} = \frac{n_{i,j}}{\sum n_{k,j}} \tag{2}$$

Where $n_{i,j}$ corresponds to the number of occurrences for each term from the article a_j and the denominator represents the occurrence of all terms in the article a_j

$$idf_i = \log \frac{|A|}{\left|\{a_j : t_i \in a_j\}\right|} \tag{3}$$

Where |A| is the total number of articles in the document collection and $\left|\{a_j : t_i \in a_j\}\right|$ is the number of articles where the term t_i appears.

Finally, a graph was constructed for each document collection, see Figure 2. Each node represents an article of a regulation text, and the associated values to the edges $V_{i,j}$ between each pair of nodes represents the inverse value of the standard similarity cosine measure.

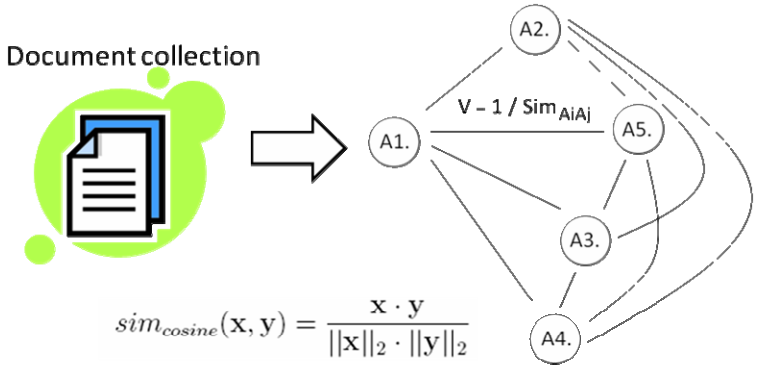


Fig. 2. Representing of the document set as a graph

3.3 Query Building Module

The question is pre-processed and then integrated into the graph in the same way that the regulation collection. Each question is converted to lowercase, punctuation symbols are eliminated (parenthesis, hyphens, numbers, etc.). Stopwords are kept (QK) or removed (QR) and finally, words that do not exist in the document collection are

removed—this is equivalent to finding a similarity measure of 0 for them. The weighting values are calculated with regard to the document collection (See eq. (2)).

3.4 Answer Extraction Module

From the query of the previous module, two new *articles* are added to the graphs (Figure 3). The answer extraction consists on finding the paths of minimal weight using the Dijkstra algorithm: Once the first minimal path is found, the nodes which constitute it are eliminated, and the Dijkstra algorithm is run again until the graph becomes disconnected for the pair of nodes which constitute the query. The *answer paths* are ordered from less to more weight, and are returned to the user with the text of each article corresponding to the node in the regulation collection.

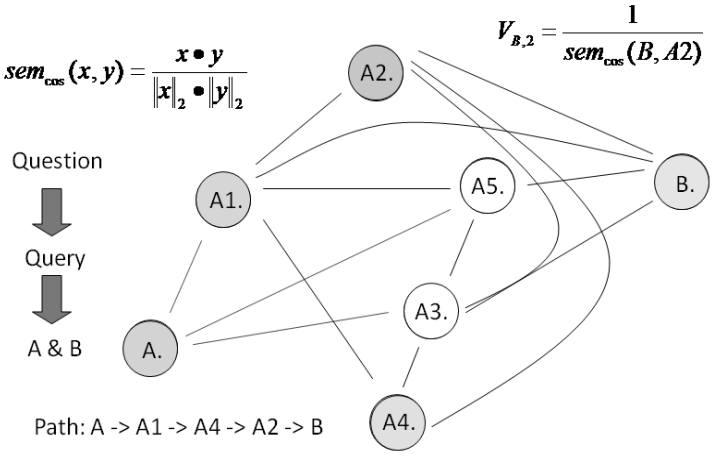


Fig. 3. A user question is split in two parts, which are added in the same way that the set of articles

4 Evaluation

The evaluation of the QAS is based on the following criteria:

i Relevance: The output of the QAS should answer to the questions of the kind "yes or no"; this is measured by determining if the articles that the general lawyer considers to produce an answers were returned by the system.

ii Noise degree in the answer: Alien and irrelevant information that the system returns is quantified.

To implement the measure of these both criteria, we use the following procedure: Initially, the answer is limited to 100 articles. It is unlikely that the answer is found after such number of articles. The returned articles were divided in groups of 5 (the maximum number of articles that can be found in a single answer of the lawyer). Each group was given the value V_i, following equation (4)

$$V_i = 1 - \frac{(n-1)}{N}, \tag{4}$$

where n is the group number (the first 5 articles constitute the group 1, the next 5, group 2, etc). N is the maximum number of packages that can be found (N=20, as 5·20=100 articles).

Finally, each article returned for a determined answer is graded using the following expression:

$$Grade \quad CR_i = \frac{\sum_{i=1}^{n} nV_i}{n_{AR}}, \tag{5}$$

where CR_i is the grade assigned to the answer of the i-th question, n is the package number where the answer-article was found, and n_{AR} is the number of answer-articles found.

The final mark of the system is the average of the grades CR_i, i.e.:

$$Grade \quad CS = \frac{\sum_{i=1}^{np} CR_i}{np}, \tag{6}$$

where np is the total number of questions for evaluation.

5 Experiments and Results

For this report, our system was tested with 21 questions. We tested against a basic Information Retrieval System (IRS) based on the vector space model. This IRS uses the same set of vectors used for the construction of the graph, but this time they were directly compared with the cosine measure with the query. The vectors which are more similar to the query are returned, so that the output of this system is the set of articles relevant to the query. The results were then compared with the results of our QAS.

We performed several experiments varying different parameters of the QAS:

a. Keeping (K) and removing (R) stopwords.
b. Different schemes of automatic question division in two parts (A and B) when converting the query.

As we mentioned in Section 3.3, we tested with keeping and removing stopwords. This yields two derived regulation document collections: DCK (Document collection keeping stopwords) and DCR (Document collection removing stopwords), and two kinds of queries, QK and QR which correspond to keeping or removing stopwords in the query.

Additionally, we performed four experiments with regard to the automatic division of the query in part A and part B. For example, consider the following question:

Cómo se lleva a cabo el procedimiento de elección de representantes alumnos ante el consejo técnico consultivo escolar? 'What is the procedure for electing the student representative at the Technical Advisory School Council?'

Table 1. Descriptions and name of each experiment. R and K refer to *Remove* or *Keep* stopwords in the document collection or the query, respectively. H and M refer to *Half division* or *Mixed division* of the query, respectively. H' and M' refer to exchanging nodes A and B in the query.

System	Document Collection	Division Type		Average Mark	Answers Found		
		Query QK	Query QR		Complete	Partial	Without ans.
QAS	**DCR**	H	-	0.4449	9	6	6
		M	-	0.4723	10	5	6
		H'	-	0.4445	10	5	6
		M'	-	0.4716	10	5	6
	DCK	H	-	0.5342	12	5	4
		M	-	0.4988	13	3	5
		H'	**-**	**0.5351**	**12**	**5**	**4**
		M'	-	0.5023	13	3	5
		-	H	0.4851	10	5	6
		-	M	0.4865	10	6	5
		-	H'	0.4858	9	6	6
		-	M'	0.4889	10	6	5
IRS	**DCK**	-	-	0.2253	3	5	13
	DCR	-	-	0.1892	4	6	11

After the procedure described in sections 3.3 and 3.4, the query is divided in two new *articles* (nodes) A and B with the following contents, according to one of the four followings types of division:

Half Division (H): Node A will contain the left half of the question, and Node B will contain the rest. If the number of words in the query is odd, the Node A will contain one word more than Node B. (*Node A*: lleva acabo procedimiento elección representantes alumnos, *Node B*: ante consejo técnico consultivo escolar)

Mixed Division (M): In this type of division, terms are mixed: odd words are in Node A and even words are in Node B. (*Node A*: lleva procedimiento representantes ante técnico escolar, *Node B*: cabo elección alumnos consejo consultivo)

Reversed Half Division (H'): As in Half Division (H) but the contents of node A and B are exchanged: (*Node A*: ante consejo técnico consultivo escolar, *Node B*: lleva cabo procedimiento elección representantes alumnos)

Reversed Mixed Division (M'): As in Mixed Division (M) but contents of the Node A and Node B are exchanged. Even words are in Node A and odd words are in Node B. *Node A*: cabo elección alumnos consejo consultivo, *Node B*: lleva procedimiento representantes ante técnico escolar)

We tested every combination of these parameters, which yielded 12 experiments. Table 1 shows the description and name of each experiment. We compared the

performance of our system with two experiments based in the document collection DCK (Document Collection Keeping stopwords) and DCR (Document Collection Removing stopwords). These experiments are shown in the last two rows of Table 1.

6 Conclusions and Future Work

Our proposed graph-based QAS has noticeably better performance with regard to the traditional model for IR; notwithstanding, the marks obtained in Table 1 show that our system still has a lot of room for improvement, considering the performed experiment which provides answers compared to those given by the general lawyer of our Institute to a set of 21 questions.

The experiments show that all marks of the QAS are between the range of 0.4445 and 0.5351. This is a very small variation, meaning that the impact of varying the division type does not have a relevant impact, *i.e.*, the arranging of the nodes has a small effect for the architecture we used.

With regard to keeping or removing stopwords, we found that keeping stopwords yields a small improvement in the performance of the system, being this only a small change.

It is important to note that these results cannot be directly compared with similar systems, because usual competitions such as TREC or CLEF consider that the answer is contained in several text fragments, whereas our system considers that answers that are not contained specifically in one or more snippets, but in their relationship—consider the example presented in Section 2: a traditional answer might be "Article 13" or "Article 43", but in this case, the answer is not found completely in one or another, but in the combination of both.

Future work includes using natural language techniques such as lemmatization and using related words from a thesaurus to smooth the similarity measures in order to improve the rank of the answer-articles.

References

1. Hirschman, L., Gaizauskas, R.: Natural Language Question Answering: The View From Here. Natural Language Engineering 7(4), 275–300 (2001)
2. Hoojung, C., Song, Y.-I., Han, K.-S., Yoon, D.-S., Lee, J.-Y., Rim, H.-C.: A Practical QA System in Restricted Domains. In: Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain, pp. 39–45 (2004)
3. Erik, T., Sang, K., Bouma, G., de Rijke, M.: Developing Offline Strategies for Answering Medical Questions. In: Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence (AAAI 2005), Pittsburgh, PA, pp. 41–45 (2005)
4. Fabio, R., Dowdall, J., Schneider, G.: Answering questions in the genomics domain. In: Proceedings of the ACL 2004 Workshop on Question Answering in Restricted Domains, Barcelona, Spain, pp. 46–53 (2004)
5. Niu, Y., Graeme, H.: Analysis of Semantic Classes in Medical Text for Question Answering. In: Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain, pp. 54–61 (2004)

6. Zhuo, Z., Sylva, L.D., Davidson, C., Lizarralde, G., Nie, J.-Y.: Domain-Specific QA for the Construction Sector. In: Workshop of IR4QA: Information Retrieval for Question Answering, 27th ACM-SIGIR, Sheffield (July 2004)
7. Paulo, Q., Rodrigues, I.P.: A question-answering system for Portuguese juridical documents. In: Proceedings of the 10th international conference on Artificial intelligence and law. International Conference on Artificial Intelligence and Law, Bologna, Italy, pp. 256–257 (2005)
8. Paulo, Q., Rodrigues, I.P.: A collaborative legal information retrieval system using dynamic logic programming. In: Proceedings of the 7th international conference on Artificial intelligence and law. International Conference on Artificial Intelligence and Law, Oslo, Norway, pp. 190–191 (1999)
9. Doan-Nguyen, H., Kosseim, L.: The problem of precision in restricted-domain question-answering. Some proposed methods of improvement. In: Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004), Barcelona, Spain, pp. 8–15.
10. Diekema Anne, R., Yilmazel, O., Liddy, E.D.: Evaluation of restricted domain question-answering systems. In: Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain, pp. 2–7 (2004)
11. Rada, M.: Random Walks on Text Structures. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 249–262. Springer, Heidelberg (2006)
12. Manning Christopher, D., Schutze, H.: Foundations of Statistical Natural Language processing. MIT Press, Cambridge (1999)
13. Salton, G., Wong, A., Yang, C.S.: A vector Space Model for Automatic Indexing. Information Retrieval and Language Processing (1975)
14. Moldovan, D., Clark, C., Harabagiu, S., Maiorano, S.: COGEX: a logic prover for question answering. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Edmonton, Canada, vol. 1, pp. 87–93 (2003)

Two Proposals of a QA Answer Extraction Module Based on Semantic Roles*

P. Moreda, H. Llorens, E. Saquete, and M. Palomar

Natural Language Processing Research Group.
Universitiy of Alicante.
Alicante, Spain
{paloma,hllorens,stela,mpalomar}@dlsi.ua.es

Abstract. The contribution of semantic roles to question answering is considered to be very valuable. Due to this fact, the aim of this paper is to analyze the influence of semantic roles in this area. In order to achieve this goal a web QA system has been implemented using two different proposals for the answer extraction module based on semantic roles, and both implementations have been evaluated for location type questions. For the first proposal, a simple set of semantic rules was created, whereas, for the second proposal, a database of possible answer semantic patterns was automatically developed. This DB is created in a first step and it will be reused each time the answer extraction module is used. Results of both approaches have been analyzed and compared showing that the patterns-based approach improves the rules-based one in precision (+ 34.40%) and recall (+ 42.80%).

Keywords: Semantic Roles, Question Answering, Semantic Rules, Semantic Patterns, Internet Search Engines.

1 Introduction

Although there are many natural languages processing (NLP) areas in which the contribution of semantic roles (SR) is considered very interesting, such as information extraction, summarization or textual entailment, several studies about automatic assignment of SR [6] indicated that one of the most outstanding contributions of SR will be in their application to question answering (QA) systems.

QA can be defined as the answering by computers to precise or arbitrary questions formulated by users in natural language instead of keyword queries or fixed templates, used in information extraction scenarios. Therefore, QA systems are one of the best examples of a new generation of search engines, providing to users exactly the required information, also in NL form [2]. The main objective of a QA system is determining "WHO did WHAT to WHOM, WHERE, WHEN, HOW and WHY?" [7].

* This paper has been partially supported by the Spanish government, projects TIN-2006-15265-C06-01 and GV06/028, and by the QALL-ME project framework, 6th Framework Research Programme of the European Union (EU), FP6-IST-033860.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 174–184, 2008.

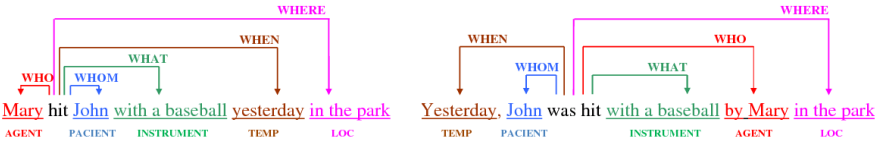


Fig. 1. Application of semantic roles in QA

SR allow to determine the semantic relations among entities and the events in which they participate, which are clearly important for text interpretation. Typical predicate semantic arguments include the role of the verb argument within the event that this verb is denoting, such as agent, patient, and instrument. Besides, semantic roles may also be found as adjuncts (e.g., locative, temporal, manner, and cause). This means that the knowledge extracted from semantic roles allows to determine, reason and represent 'WHO did WHAT to WHOM, WHERE, WHEN, HOW and WHY?" in a sentence, and therefore, give answers to questions about these sentences, as shown in figure 1.

Despite the fact that some proposals according to this line of research have been previously formulated, as explained in next sections, a depth analysis of the influence of semantic roles in QA systems is still necessary, being the objective of this paper.

The paper is structured as follows: Section 2 exposes the background of SR field applied to QA systems, Section 3 provides detailed information of our two proposals for an answer extraction module based on semantic roles in a general purpose QA system, and Section 4 includes the evaluation and a discussion of the results for both proposals. Finally some conclusions and orientations for future work are presented.

2 Background of Semantic Roles Applied to QA

Since the first automatic semantic role labeling (SRL) system [6], the application of SR to QA systems was presented as a proposal. However, not before 2004, this proposal was implemented and some results were presented.

All the QA systems based on SR and, in general all the QA systems, have a very similar architecture, and as described in this field literature [2], this general architecture is summarized in the following modules: 1) Question analysis that extracts all the useful information from the question, 2) Document retrieval to obtain a set of relevant documents, 3) Passage retrieval to obtain only the relevant information from retrieved documents and 4) Answer extraction that determines which parts of the selected passages are potential answers.

In order to study the influence of semantic roles in QA, the used role set and the way semantic roles have been used in the QA system has to be analyzed.

The SR sets that most systems have used are: the set used in the corpus developed in PropBank (PB), the Proposition Bank project [18], such as [21,22,10];

the set developed in the FrameNet (FN) project [3], such as [17,20,5]; or a combination of both [15,8]. Other systems have selected a subset of the role sets given by these corpus, such as [9], which is only using PB roles A0, A1 and A2.

Regarding the use of the semantic roles in the QA systems, systems can be divided into two main groups: a) the use of semantic roles to complement other methods, and b) the use of semantic roles as a core method of a module in the QA architecture.

2.1 Roles as a Complementary Method

In this case, QA systems are based on named entity recognition (NER), and the use of SR is only providing additional information in order to analyze the possible improvement in the results of the QA system [22,9,20,10].

These types of approaches are only giving information about how semantic roles are able or not to complement a NER approach. Then, it is impossible to know how SR are able or not to improve QA systems themselves.

2.2 Roles as a Core Method

These approaches are using semantic roles to perform one module of the QA system. A brief summary of the main systems is presented in table 1.

As shown in the table, most of the systems are using a mapping between the semantic information of the question and the semantic information of candidate answers. Narayanan and Harabagiu's [15] was one of the first proposal about using SR in QA systems and they were applied to determine the type of the answer of complex questions. Their evaluation results over an ad-hoc set of 400 questions indicated a 73.5% of correct answer types were properly detected. Ofoghi et al. [16] implemented a manual proof over a set of 15 questions in order to extract candidate answers to a question using semantic roles. The evaluation of this approach using TREC2004 question corpus showed an MRR of 38.89 %. Kaisser's et al. [8] is a very similar proposal to the explained before. This system looks for the absent role in the question into the roles of the candidate answers. This system was evaluated with a subset of TREC2002 question corpus and obtained a accuracy of 36.70%. Fliedner [4] proposes a representation of both question and passages containing a possible answer as FrameNet style structures. The answer is obtained by a mapping process between both structures. Results for open domain questions achieved a precision of 66% and a 33% in recall.

Table 1. Summary of the use of semantic roles in QA systems

System	Roles	Use	Method
Narayanan	FN	QC	Mapping Q. Pattern – Answer Pattern
Stenchikova	PB	AE	Rules type Q. – Answer role
Ofoghi	FN	AE	Mapping Q. Pattern – Answer Pattern
Kaisser	FN,PB	AE	Mapping Q. Pattern – Answer Pattern
Moschitti	PB	QC, AC, AR	Supervised Machine Learning
Fliedner	FN	AE	Mapping Q. *Frame* – Ans. *Frame*

Besides, another system like Stenchikova [21] is establishing a set of rules that relate some types of questions (who,when,where or what) with the role type for the expected answer. In this case, the evaluation of the system obtains an MRR of 35%. Otherwise, Moschitti [14] proposes a supervised learning algorithm using information of semantic analysis tree composed of the sentence predicate and its arguments tagged with SR. Results obtained prove the usefulness of this information for classification (MRR 56.21%) and re-classification (MRR 81.12%) of answers, but not for the question classification. One of the most important problems of all these systems is the extraction of the semantic roles of the question. This is due to the fact that the semantic role labeling tools have serious problems to annotate questions.

Once the different proposals have been analyzed, it seems obvious that the main contribution of SR to QA systems is in the answer extraction module. However, previous works are not comparable between them due to the fact that these systems are using different set of measures, questions, documents and roles, to achieve different objectives: determine the type of the answer and extract a list of possible answers.

3 Two Proposals of Answer Extraction Based on Semantic Roles

With the main objective of qualitatively determining the influence of SR in QA systems, this paper analyzes the behavior of the answer extraction module regarding: a) rules that establish relationships between the type of questions and SR and, b) patterns that have been built using the information of SR.

Previous to this analysis, the following decisions have been taken:

- *Role set.* Due to the problem of adjuncts coverage of FN, we decided to use the semantic role set provided by PB.
- *Question Answering system.* A QA system has been developed following the steps indicated in Pizzato et al. [19]. This system is made up of the general modules of a QA system explained in Section 2. In our case, the information retrieval module uses snippets obtained from several Internet search engines. Moreover, the answer extraction module has been modified in order to apply SR approaches.

Taking into account that the behavior of QA systems could be different depending on the kind of SR, this work analyzes only questions that can be answered with a location expression. This specialization does not mean that the same analysis can not be made about other kind of questions. In fact, it is only necessary to repeat this process for each type of question.

3.1 Answer Extraction Based on Rules

Depending on the kind of question, and therefore the expected answer, a different set of SR could be considered as a possible answer. So, it is possible to

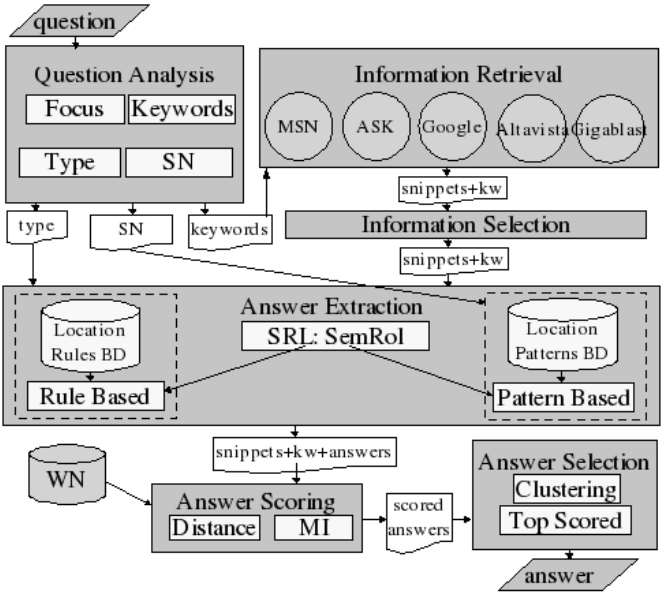


Fig. 2. Architecture of the QA system using SR in answer extraction

define a set of semantic rules that establishes relationships between the type of the question and a SR. For instance, questions such as "Where", "In where + location expression", "In what + location expression" or "What + location expression" must be answered with the location semantic role and not with the agent, patient, temporal, cause or mode semantic role. A summary of these semantic relationships is shown in table 2 [12].

Making use of these rules, only sentences containing the right semantic roles are selected. So, the answer extraction module will select the arguments of the snippets returned by the information retrieval module, that play the location role. SemRol tool [13] will be used in order to determine the role of the arguments.

Such rules avoid the necessity of annotating the questions with semantic roles. Therefore, the problems found in other QA systems based on rules, that have to annotate questions [21], disappear.

3.2 Answer Extraction Based on Patterns

The set of roles used in PropBank prevents making consistent generalizations about adjunct or numbered arguments [18]. Although the set of roles is unique for all verbs, it is also variable for each verb and its senses. Then, the rules defined in section 3.1 not always work in the right way.

For instance, the sentences shown in (example 1) and (example 2) have an argument with the location role, "to the John's house" and "to the park", respectively. However, neither of both has the AM-LOC role. Indeed, in one case,

Table 2. Question-SR semantic relation set (R: Semantic Rules and P: Semantic Patterns)

Q. Type	Role	PB Role	Role No
LOCATION	Location	AM-LOC	ProtoAgent
		A2	Mode
		A3	Temporal
		A4	Cause
			ProtoPatient
TEMPORAL	Temporal	AM-TMP	ProtoAgent
			Mode
			Location
			Cause
			ProtoPatient
MANNER	Mode	AM-MNR	ProtoAgent
	Theme (if it is a diction verb)	A1	Location
			Temporal
			Cause
			Patient
			Beneficiary
HUMAN (Who)	Agent – ProtoAgent	A0	Mode
	Patient – ProtoPatient	A1	Temporal
			Location
			Theme
			Beneficiary
HUMAN (Whose)	Receiver	A2	Agent
	Beneficiary	A3	Location
	Patient		Mode
	ProtoPatient		Temporal
			Theme
			Cause
REASON	Cause	AM-CAU	
	Theme	A1	

the location role is represented by the A2 role (example 1), and in the other, by the A4 role (example 2).

Example 1. $[_{A0}$ Mary] is going $[_{A2}$ to the John's house].

Example 2. $[_{A0}$ Mary] is going $[_{A4}$ to the park].

Such as [12] showed, in PropBank, the location can be represented by the A2, A3, A4 or AM-LOC semantic roles. So, the answer extraction module based on rules is not able to detect all the possibilities. A first idea could be consider the AM-LOC when appears, and the other roles when not. This is possible because when the A2, A3 or A4 roles represent location, no other argument can have the location role. Therefore, it is impossible that the AM-LOC role appears. The problem is determining which of the roles, A2, A3 or A4, represent the location role if they appear in the same sentence and determining when A2, A3 or A4 role represent location.

To solve this problem, and considering the work presented in [23] about an answer extraction module based on patterns using named entities, the automatic construction of a set of semantic patterns based on semantic roles is proposed. This set of semantic patterns will cover a lot of possibilities in which semantic roles represent location.

The automatic construction of patterns based on SR can provide two important profits. On one hand, an argument with a specific role in a sentence has always the same role regardless the position, tense or voice of that sentence. For instance, in the figure 1, "yesterday" has the temporal role in both sentences. On the other hand, with an automatic process, it is possible to obtain much more patterns than with a manual process, and with a lower cost. Therefore, the two most important problems that could present the use of patterns are solved. Besides, this process does not need to label the semantic roles of the question. Then the problems presented in systems shown in section 2.2 are resolved.

So, following the work of [23] and using pairs of question-answer, the automatic construction of a set of semantic patterns based on semantic roles is proposed. This process consist of four stages:

1. *Snippet retrieval.* For each pair question-answer, the set of terms which a relevant document should contain, is defined. Then, a query using these terms is submitted to the Web and the snippets retrieved containing some of the terms are selected.
   (a) The set of relevant terms is composed of 1) the noun phrases of the question, and 2) all the possible combinations of sub-phrases of the answer.
   (b) The search engines used to submit the terms to the Web are MSN, AskJeeves, Google, Altavista and Gigablast[1].
   (c) The first 100 snippets retrieved for each search engine containing the terms of the question and at least, one of the terms of the answer, in the same sentence, are selected.
2. *Semantic Filtering of snippets.* The snippets selected are validated according to the semantic relation that they contain. Since the semantic relation generally appears as the main verb of the question, all verbs in the selected snippets are examined. Sentences of snippets containing synonyms, hyperonyms or hyponyms of the question verb, are selected. This semantic information is obtained from WordNet [11].
3. *Generating the answer pattern.* Finally, the selected sentences are generalized in semantic patterns using information about semantic roles.
   (a) Each sentence is annotated with semantic roles using the SemRol tool[13] in order to identify location arguments (AM-LOC, A2, A3 or A4 semantic roles).
   (b) Then, the argument corresponding to some of the sub-phrases of the answer are replaced by its semantic role tag.
   (c) Next, the arguments corresponding to the noun phrases of the question are replaced by $< QARG_n >$ tags, where n is the phrase counter.
   (d) Finally, the other arguments of the sentence are replaced by $< ARG_n >$ tags, where n is the argument counter.
   (e) All other words are removed.
4. *Eliminating duplicates.* Regardless the position of the tags, if two patterns have the same tags but different verbs, a single pattern is obtained containing the set of tags and a list of those verbs.

[1] All of them consulted in March 2008.

An example pattern is shown in example 3.

Example 3. $be, bear, \mid < QARG_1 >< AM - LOC >< ARG_1 > \mid 302$

After this previous process to automatically obtain a database of semantic patterns (only performed once), the answer extraction module of the QA system is modified in the following way: when a new location question is formulated, one or more patterns (one for each location semantic role AM-LOC, A2, A3, A4 in the sentence) for the returned snippets of this question are obtained and they are matched with the set of patterns in our database. If there is a coincidence, the text corresponding to the semantic role tag in the pattern (i.e. LOC in example 3) is retrieved as an answer. To do this, the sentences of the snippets are annotated with semantic roles, using the SemRol tool[13], and they are generalized in patterns.

4 Evaluation and Results Discussion

4.1 Evaluation Environment

A set of 100 questions, built using a subset of TREC1999 and TREC2000 factoid location questions and answers, has been used for testing.

Before carrying out the test, a Patterns database (DB) for pattern-based answer extraction module has to be built, as explained above. It has been built using a set of 200 questions, composed of a subset of TREC2003, TREC2006 and OpenTrivia.com factoid location questions and answers.

As explained in section 3, our system uses internet search engines results as corpus to answer the questions. We judged answers to be correct if they represent or contain the correct answer. The measures used to evaluate the system are precision (*questions answered correctly/total questions answered*), recall (*questions answered correctly/total questions*)and F1 (*(2*Precision*Recall)/(Precision+ Recall)*).

4.2 Results Analysis

The QA system is executed for both implemented answer extraction modules and precision, recall and F1 measures are calculated. It must be highlighted that neither manual review of sub-processes outputs nor post-execution adjustments have been made to automatic processes of presented system.

Table 3 shows the results obtained in the evaluation of both approaches and the pattern-based improvement percentage.

On one hand, regarding the aim of evaluating the influence of SR in QA task, and specifically the differences between the two described answer extraction modules, a comparative analysis has been made. Two main conclusions can be extracted from the results:

– Patterns-based approach improves rules-based in recall because of the inclusion of A2, A3 and A4 roles as possible answers for some patterns. One

Table 3. Results for the QA system with SR-based answer extraction module

	Rules (%)	Patterns (%)	Improv. (%)
Precision	65.60	88.20	34.40
Recall	21.00	30.00	42.80
F1	31.80	44.80	40.80

of the verbs that takes profit of this fact is "to come" whose A2 argument involves normally a location like "from somewhere". Moreover, considering synonym, hyperonyms or hyponyms verbs in Patterns DB building process widens the approach coverage by including more than only questions verbs.
- Patterns improve rules in precision because while rules-based module extracts all location roles as possible answers, patterns-based only considers location roles whose pattern represent one of the contained in Patterns DB. That way, patterns-based approach does a kind of semantic filtering of sentences that results in a more precise extraction of answers.

On the other hand, the recall achieved is due to type of questions and the null manual adjustment of the system. As an overall trend we can notice that most of either TREC and Trivial questions are built to be solved by Named Entities instead of Semantic Roles because of both questions and answers are mostly composed by proper nouns. Nevertheless, if we consider questions whose answer is a common name (i.e.: Where does chocolate comes from? Cocoa bean), QA systems based only on NEs will not be able to find the answer because it (i.e.: Cocoa bean) will not be a NE. However, for a system based on semantic roles it would be very simple, by finding the argument of the verb with the location role.

Furthermore, it is important to remark that the presented system is completely built up of automatic processes executed without any manual review.

This results are not compared with other systems results because each system is using different set of measures, questions, documents and roles, to achieve different objectives: determine the type of the answer and extract a list of possible answers.

5 Conclusions and Further Work

The aim of this paper is to analyze the influence of using semantic roles in question answering systems. It do not pretend neither creating the best QA system nor comparing SR question answering systems with NE ones. To reach this goal a QA system was developed and two proposals of a QA answer extraction module based on semantic roles have been implemented. In the first proposal a set of semantic rules that establishes relationships between the type of the question and a SR is defined. In the second one, the automatic construction of a set of semantic patterns of possible answers based on semantic roles is proposed. This set of semantic patterns will cover a lot of possibilities in which semantic roles represent location.

Both proposals have been evaluated for location type questions. On one hand, results from the evaluation show that patterns-based approach improves

rules-based in both precision (+34.40%), recall (+42.80%) and F1 (+40.80%). It is necessary to consider that the achieved recall is this is due to the fact that questions normally used to evaluate QA systems are mostly oriented to be solved by named entities and not by semantic roles, as well as there are not manual adjustments in presented system.

As further work, improvements in two different lines have been proposed:

- Improving use of SR: a) Extending the QA system and SR-based extraction modules to other type of questions such as Person, Organization or Time-Date, and b) creating a question set with answers based on common names, to use it together with common question sets (i.e.: TREC question sets) mainly based on proper names to compare SR with NE answer extraction approximations in equitable conditions.
- Improving the QA system, in order to deeper analyze semantic roles impact: a) Implementing an Answer Clustering module based on semantic information, and b) adjusting answer scoring and ranking module to better select the final answer among the candidate answers.

References

1. Deep Linguistic Processing Workshop in 45th Annual Meeting of the ACL, Prague, Czech Republic (June 2007)
2. Ferrández, A.: Sistemas de pregunta y respuesta. Technical report, Universidad de Alicante (2003)
3. Fillmore, C.J.: Framenet and the linking between semantic and syntactic relations. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING), Taiwan, pp. xxviii–xxxvi (2002)
4. Fliedner, G.: Linguistically Informed Question Answering. Saarbrücken Dissertations in Computational Linguistic and Language Technology, vol. XXIII. Universität des Saarlandes und DFKI GmbH, Saarbrücken (2007)
5. Frank, A., Krieger, H., Xu, F., Uszkoreit, H., Crysmann, B., Jorg, B., Schafer, U.: Question answering from structured knowledge sources. Journal of Applied Logic. Special issue on Questions and Answers: Theoretical and Applied Perspectives 5(1), 20–48 (2007)
6. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Computational Linguistics 28(3), 245–288 (2002)
7. Hacioglu, K., Ward, W.: Target Word Detection and Semantic Role Chunking Using Support Vector Machines. In: Proceedings of the Human Language Technology Conference (HLT-NAACL), Edmonton, Canada (June 2003)
8. Kaisser, M.: Question Answering based on Semantic Roles. In: Proceedings of the Deep Linguistic Processing Workshop in 45th Annual Meeting of the ACL, [1]
9. Lo, K.K., Lam, W.: Using semantic relations with world knowledge for question answering. In: Proceedings of The 15th Text Retrieval Conference (TREC) (2006)
10. Melli, G., Wang, Y., Liu, Y., Kashani, M.M., Shi, Z., Gu, B., Sarkar, A., Popowich, F.: Description of SQUASH, the SFU Question Answering Summary Handler for the DUC-2005 Summarization Task. In: Proceedings of the Document Understanding Conference 2006 (DUC 2006), New York City (June 2006)

11. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Five Papers on WordNet. CSL Report 43. Technical report, Cognitive Science Laboratory, Princeton University (1990)
12. Moreda, P., Navarro, B., Palomar, M.: Corpus-based semantic role approach in information retrieval. Data and Knowledge Engineering 61(3), 467–483 (2007)
13. Moreda, P., Palomar, M.: The Role of Verb Sense Disambiguation in Semantic Role Labeling. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) FinTAL 2006. LNCS (LNAI), vol. 4139, pp. 684–695. Springer, Heidelberg (2006)
14. Moschitti, A., Quarteroni, S., Basili, R., Manandhar, S.: Exploiting Syntactic and Shallow Semantic Kernels for Question Answer Classification. In: Proceedings of the Deep Linguistic Processing Workshop in 45th Annual Meeting of the ACL, [1], pp. 776–783.
15. Narayanan, S., Harabagiu, S.: Question answering based on semantic structures. In: Proceedings of the 20th International Conference on Computational Linguistics (COLING), Switzerland (August 2004)
16. Ofoghi, B., Yearwood, J., Ghosh, R.: A hybrid question answering schema using encapsulated semantics in lexical resources. In: Advances in Artificial Intelligence, 19th Australian Joint Conference on Artificial Intelligence, Australia (2006)
17. Ofoghi, B., Yearwood, J., Ghosh, R.: A semantic approach to boost passage retrieval effectiveness for question answering. In: Computer Science 2006, 29th Australian Computer Science Conference, Australia, pp. 95–101 (2006)
18. Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: An Annotated Corpus of Semantic Roles. Computational Linguistics 31(1), 71–106 (2005)
19. Sangoi Pizzato, L.A., Mollá-Aliod, D.: Extracting Exact Answers using a Meta Question answering System. In: Proceedings of the Australasian Language Technology Workshop 2005 (ALTW 2005), Sidney, Australia (December 2001)
20. Shen, D., Wiegand, M., Merkel, A., Kazalski, S., Hunsicker, S., Leidner, J.L., Klakow, D.: The Alyssa System at TREC QA 2007: Do We Need Blog06? In: Proceedings of The 16th Text Retrieval Conference (TREC), USA (2007)
21. Stenchikova, S., Hakkani-Tur, D., Tur, G.: Qasr: Question answering using semantic role for speech interface. In: Proceedings of the International Conference on Spoken Language Processing, Pittsburg (2006)
22. Sun, R., Jiang, J., Tan, Y.F., Cui, H., Chua, T., Kan, M.: Using Syntactic and Semantic Relation Analysis in Question Answering. In: Proceedings of The Fourteenth Text Retrieval Conference (TREC 2005) (2005)
23. Yousefi, J., Kosseim, L.: Using semantic constraints to improve question answering. In: Kop, C., Fliedl, G., Mayr, H.C., Métais, E. (eds.) NLDB 2006. LNCS, vol. 3999, pp. 118–128. Springer, Heidelberg (2006)

A Misclassification Reduction Approach for Automatic Call Routing

Fernando Uceda-Ponga[1], Luis Villaseñor-Pineda[1], Manuel Montes-y-Gómez[1], and Alejandro Barbosa[2]

[1] Laboratorio de Tecnologías del Lenguaje, INAOE, México
{fuceda,villasen,mmontesg}@inaoep.mx
[2] Nuance Technologies, Mexico
Alejandro.Barbosa@nuance.com

Abstract. Automatic call routing is one of the most important issues in the call center domain. It can be modeled –once performed the speech recognition of utterances– as a text classification task. Nevertheless, in this case, texts are extremely small (just a few words) and there are a great number of narrow call-type classes. In this paper, we propose a text classification method specially suited to work on this scenario. This method considers a new weighting scheme of terms and uses a multiple stage classification approach with the aim of balance the rate of rejected calls (directed to a human operator) and the classification accuracy. The proposed method was evaluated on a Spanish corpus consisting of 24,638 call utterances achieving outstanding results: 95.5% of classification accuracy with a rejection rate of just 8.2%.

1 Introduction

Call routing (CR) is one of the most important issues in the call center domain. It can be defined as the process of associating user requests with their desire destinations [1]. In other words, it involves forwarding incoming phone calls to someone qualified to handle them (e.g., to the most appropriate customer representative).

Traditionally, this task has been performed using a touch-tone approach, which allows users to navigate through different options by a hierarchical menu. In spite of its conceptual simplicity, this approach has several disadvantages, such as the time the users need for listening all options as well as the difficulty for matching existing options with their specific needs. These inconveniences are evident even for simple requests such as "I want to know my account balance", which may require users to navigate as many as four or five nested menus with four or five options each [2].

The inconveniences of the tone-touch approach leaded to the emergence of *automatic call routing systems*, which allow users to interact in natural spoken language [1], and therefore, help companies to significantly reduce their telephone time costs. Typically, this kind of systems considers two main steps. In the first one, phone calls are transcribed to text by means of a speech recognition process. Then, in a second step, text transcriptions are used to predict the correct destination for the phone calls. In particular, this paper focuses on the second step, which, in fact, can be considered as a *text classification* problem.

Text classification, the assignment of free text documents to one or more predefined categories based on their content, is a widely studied problem that has recently

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 185–192, 2008.

achieved significant advances [3]. Nevertheless, it is important to point out that automatic call routing is a more difficult problem than ordinary text classification, since phone call transcriptions are very short and noisy texts. That is, they are texts consisting of just a few words and containing many errors such as word insertions, deletions and substitutions.

In addition, automatic call routing differs from conventional text classification in that, in the former, there is always the possibility of appealing for human assistance. In particular, call routing systems tend to *reject* all ambiguous or incomprehensible transcriptions, transferring their resolution to a human operator. Evidently, this circumstance is of great relevance, since having a high rejection rate implies a high telephone time cost, whereas having a low classification accuracy (due to the misclassification of unclear transcriptions) causes a strong discomfort among users. Therefore, one of the main research topics on automatic call routing is to provide a good balance between these two factors, the classification accuracy and the rejection rate.

In this paper, we propose a classification method specially suited to the task of automatic call routing. On the one hand, this method considers a new *weighting scheme* that is especially appropriate for short texts. On the other hand, it uses a *two-step classification approach* that allows achieving an adequate balance between the rejection rate and the classification accuracy. In a first step, this approach guarantees high classification accuracy by tolerating a high rejection rate. Subsequently, in a second step, it reclassify rejected transcriptions with the intention of recovering some instances and, therefore, reducing the final rejection rate.

Experimental results on a Spanish corpus consisting of 24,638 call utterances are encouraging; the proposed method could correctly classified 95.5% of the utterances with a rejection rate of just 8.2%, outperforming the application of other text classification approaches.

The rest of the paper is organized as follows. Section 2 describes some previous work on automatic call routing. Section 3 describes the proposed method for automatic call routing. Section 4 presents the evaluation results. Finally, Section 5 gives our conclusions and describes some ideas for future work.

2 Related Work

There exist different approaches for automatic call routing; among them we can mention the following three ones:

1. A straightforward approach based on the use of *keywords* for triggering different destinations [4].
2. An approach based on the use of *language models* that describe the main characteristics (words sequences) of each destination [5].
3. An approach based on the application of traditional *text classification* methods [2,6].

From these approaches, the first one is the simplest to understand and implement; nevertheless, it is very sensible to transcription errors. The second approach is more robust than the former, but it requires more “training data” as well as clearly differentiable destination vocabularies. Finally, the third approach is less sensible to transcription errors, but it is damaged by the short length of transcriptions as well as by the great number of narrow destinations (categories).

Different works have evaluated the effectiveness of conventional text classification methods for automatic call routing. For instance, [7] shows a comparison of several learning algorithms on a corpus of 4000 manual transcriptions corresponding to nine –balanced– categories. It reports a result of 81.6% of accuracy using a simple recurrent network. In the same way, [8] reports an accuracy of 75.9%; nevertheless, in this case they used a small corpus of 743 automatic transcriptions from six different categories. Both works do not report the rejection rate.

More recently, some works have proposed different adaptations to the traditional text classification approach. These adaptations consider the use of different utterance characterizations, weighting schemes, feature selection methods and learning strategies. For instance, [9] presents a learning approach that uses a cascade of binary classifiers (of Support Vector Machines) with the aim of reducing the classification errors. In this approach, those instances that could not be classified (i.e., that pass through all classifiers) are rejected. Reported results on a set of automatic transcriptions from 19 categories are very relevant; on the one hand, it reports a best accuracy of 99% corresponding to a rejection rate of 75%, and on the other hand, a best rejection rate of 15% for a classification accuracy of 95%.

Similar to these works, our automatic call routing method is also based on a supervised learning approach. Its main difference relays on the use of a novel two-step classification approach that allows maintaining a good balance between the accuracy and rejection rates. In addition, it applies a weighting scheme that is less sensible to the short length of texts. The following section introduces the proposed method.

3 Proposed Method

As we previously mentioned, our method includes two major modifications to the traditional text classification approach. On the one hand, it uses a new term weighting scheme that allows improving the discrimination among narrow classes formed by very short texts. On the other hand, it considers a two-step classification approach that helps maintaining an adequate balance between the accuracy and rejection rates. In the following subsections, we describe in detail these two modifications.

3.1 Term Weighting Scheme

Typically, term relevance is determined using the well-known *tf-idf* weighting scheme. This scheme, however, presents some drawbacks for its application to automatic call routing. First, given the short length of texts, the *tf*-value is quite similar for most of the terms from a category[1]. Second, given the presence of several related narrow categories, the *idf*-value is also very similar for most terms. Therefore, this weighting scheme does not allow achieving a good discrimination among categories.

Based on the previous observations, we propose a weighting scheme that computes the weight of terms in relation to their occurrence in the whole categories instead than in individual utterances. This scheme considers that the probability of occurrence of a term across distinct categories is different. In particular, we compute the weight of a term t in a category $c_i \in C$ as follows:

[1] This is because users tend to use a simple and direct language (more restricted vocabulary) when they interact with a machine (an automatic system).

$$w_t^{c_i} = \frac{p_t^{c_i}}{\max_{\forall k \in c_i}\left(p_k^{c_i}\right)}$$
$$p_t^{c_i} = P(c_i \mid t) = \frac{f_t^{c_i}}{\sum_{\forall c_k \in C} f_t^{c_k}} \quad (1)$$

where C is the set of categories, $f_t^{c_i}$ is the frequency of occurrence of the term t in the category c_i, and $p_t^{c_i}$ indicates the conditional probability $P(c_i|t)$, that is, the probability of having the category c_i given the presence of the term t.

It is interesting to notice that the final weight of a term in a category is obtained by applying a kind of normalization with respect to the probabilities of other terms from the same category. Roughly speaking, this normalization allows giving greater weight to terms clearly related to the category and reducing the value of terms uniformly distributed among several classes.

3.2 Two-Step Classification Approach

Figure 1 shows the general architecture of the proposed classification approach, which consists of two main steps: a high-precision classification step and a misclassification reduction step. The main purpose of this new approach is to achieve a *good balance between the classification accuracy and the rejection rate*.

The first step achieves an initial classification of the input utterances (i.e., the phone call transcriptions). This classification considers all categories including an "unknown category" that gathers all ambiguous and incomprehensible transcriptions. This category functions as a rejection class, since its elements are transferred to a human operator for their resolution. This way, the purpose of this step is to guarantee a high classification accuracy by tolerating a high rejection rate. In other words, in this step only the most confident utterances are sent to content categories, whereas the rest of them are classified as unknown.

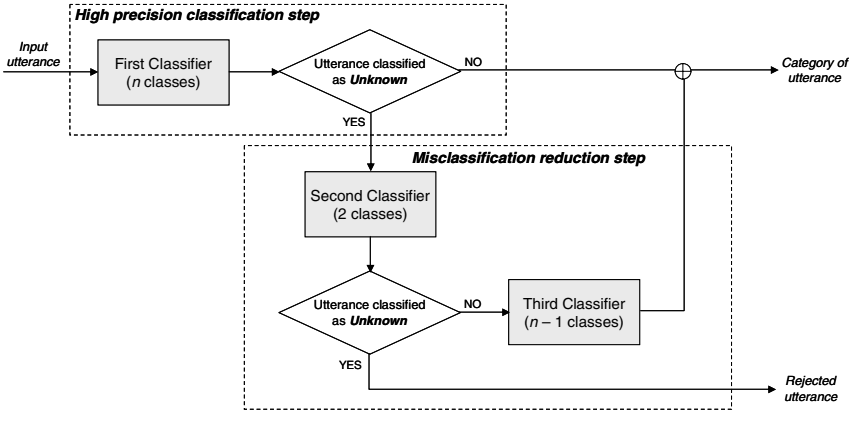


Fig. 1. General architecture of the proposed method

Subsequently, in a second step, the idea is to reclassify the set of rejected transcriptions with the intention of recovering some utterances and, therefore, reducing the final rejection rate. This step combines two additional classifiers. One of them considers only two categories (unknown and the rest), and it is especially suited to distinguish clear ambiguous and incomprehensible transcriptions. The other one considers all categories except unknown and its purpose is to reallocate rescue utterances within content categories.

It is important to point out that the weight of terms (defined in accordance to formula 1) varies from one classifier to another because they consider different sets of categories. This characteristic of our approach is very relevant, since it allows giving the terms a convenient (highly discriminative) weight at each step.

4 Experimental Evaluation

4.1 Train and Test Corpora

In order to evaluate the proposed method we used a corpus consisting of 24,638 Spanish automatic transcriptions[2]. These transcriptions came from a company that provides TV, internet and telephone services, and were manually classified in 24 different categories. The first 23 categories correspond to content categories (i.e., to possible destinations for user requests), whereas the last one is an "unknown category", which gathers all utterances that could not be classified and that must be rejected.

For evaluation purposes, we divided the corpus in two non-overlapped subsets: a train set formed by 80% of the instances from each category (in total, 19,694 utterances), and a test set including the remaining 20% of the instances (4,944 utterances). Figure 2 shows the distribution of instances per category; as it can be observed, this is a very imbalanced corpus.

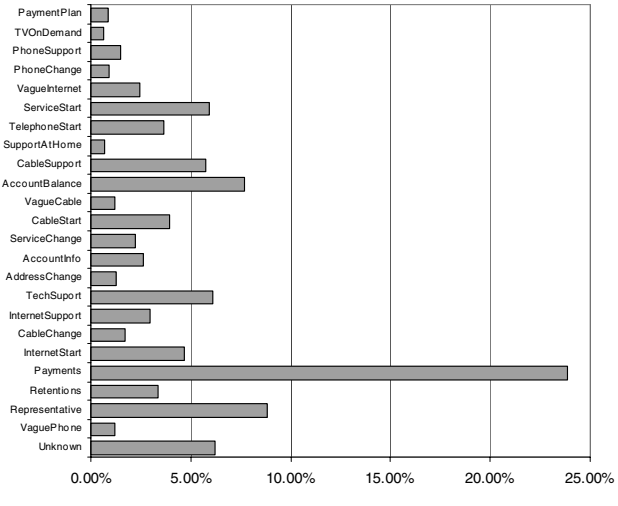


Fig. 2. Percentage of instances per category in the used corpus

[2] It was assembled by Nuance technologies (*www.nuance.com*) using the Wizard of Oz technique.

We decided using this evaluation scheme because we wished to simulate a real situation, where it is possible to find utterances containing unspecified words for the classifier. In particular, the vocabularies from the train and test sets showed important differences: 13.5% of the words from the test instances were not included in the train set.

4.2 Results

In order to obtain a general vision from the proposed method we performed several experiments. Initially, we studied the usefulness of the weighting scheme for tuning the accuracy and rejection rates. Then, we evaluated the effectiveness of our two-step classification approach by comparing its results against others from conventional one-step classifiers.

Tuning the accuracy and rejection rates
For this first experiment, we used the weight of terms as the main criterion for *feature selection*. In particular, we eliminated terms with weight –in all categories– less than a given specified threshold. Table 1 presents the results corresponding to different thresholds. It mainly shows the number of training features, the classification accuracy[3] as well as the rejection rate[4]. It is important to mention that in all cases we used a single (*one-step*) multi-class Naïve Bayes classifier and applied a Boolean weighting for the test utterances[5].

Table 1. Feature selection using the proposed weighting scheme

Weight threshold	Number of features	Classification accuracy	Rejection rate
0	2474	64.9%	1.0%
0.1	2473	71.6%	0.9%
0.2	2431	87.5%	24.2%
0.3	2290	97.4%	31.5%
0.4	2074	99.5%	40.7%
0.5	1712	99.9%	50.9%

Results from this initial experiment showed that using different thresholds it was possible to obtain different combinations of classification accuracies and rejection rates. As we expected, the use of the most discriminative terms allowed achieving better classification accuracy, but also incremented the rejection rate since several instances could not be represented. These results are of great relevance to our method since it considers three different classifiers having different purposes.

Evaluating the effectiveness of our two-step classification method
The goal of the proposed two-step classification method is to achieve a good balance between the classification accuracy and the rejection rate. In order to evaluate the effectiveness of this method, we compared its results against those from a conventional one-step classifier (refer to Table 1).

[3] Defined as the percentage of utterances correctly classified within the set of 23 content categories.
[4] Defined as the percentage of test utterances classified as unknown.
[5] In contrast, training instances were represented using the proposed weighting scheme.

Table 2. Results from the proposed two-step classification method

Correctly classified Utterances		Classification accuracy	Rejection rate
First step	*Second step*		
3396	938	95.5%	8.2%

Our two-step classification method, as shown in Figure 1, considers the combination of three different classifiers. All of them, as in the previous experiment, were implemented using a Naïve Bayes classifier and Boolean weights for representing test instances. Nevertheless, in this case, each classifier considers a different weight threshold[6]: the first two classifiers used a threshold of 0.3, whereas the last one used a threshold of 0.2.

Table 2 shows the results achieved by this method. In this case, the accuracy was calculated taking into consideration the correctly classified instances from the first step as well as the correctly rescued instances from the second step. On the other hand, the rejection rate was obtained from the set of instances classified as unknown by the second classifier. As can be observed, the achieved results are encouraging; the proposed method reached an adequate balance between the classification accuracy and rejection rate, better than the results from all previous one-step classifiers (refer to Table 1).

Finally, in order to have more information to judge the effectiveness of the proposed method, we compared its results against those from the AdaBoost algorithm (commonly used in this task [10]). Table 3 shows the results from this comparison. The first row presents the results of our method, and the second and the third rows show the results from AdaBoost using Naïve Bayes as base classifier and appliying six iterations. The only difference from these two runs was the used weighting scheme. The second row reports the result using the proposed weighting scheme, whereas the last row shows the result corresponding to the *tf-idf* weighting scheme.

Table 3. The proposed method against the AdaBoost classifier

Classification scheme	Classification accuracy	Rejection rate	Recall of *unknown* class
Our method	95.5%	8.2%	100%
AdaBoost (using our weighting scheme)	98.9%	29.7%	100%
AdaBoost (using *tf-idf* weights)	78.2%	6.8%	60%

The results from this experiment are very interesting; they show that our method allowed achieving a better balance between the accuracy and rejection rates than the traditional AdaBoost method. On the other hand, they also show that the proposed weighting scheme lead to better results than the *tf-idf* weighting scheme. Moreover, using the *tf-idf* weighting scheme it was possible to obtain a small rejection rate, however, it could not identify (and therefore, transfer to a human operator) all ambiguous or incomprehensible utterances. This last factor is very relevant since the misclassification of unclear transcriptions tend to cause a strong discomfort among users.

[6] These thresholds were empirically determined; for each classifier we evaluated several thresholds and selected the combination that achieved the best overall accuracy.

5 Conclusions

This paper proposed a new text classification method that is specially suited to the task of automatic call routing. This method differs from previous works in two main concerns: the application of a new term weighting scheme, and the use of a two-step classification approach.

Experimental results on a Spanish corpus consisting of 24,638 call utterances showed, on the one hand, that the proposed weighting scheme is especially appropriate for short texts, and on the other hand, that the two-step classification approach allows maintaining an adequate balance between the classification accuracy and the rejection rate. In other words, these results indicated that the proposed method is pertinent for the automatic call routing task, which commonly considers a great number of narrow categories, imbalanced training data sets as well as very short text instances.

Another important aspect is that the method itself does not rely on language dependent rules, and therefore, it would be easily used in other languages.

Finally, it is important to mention that the method can be adjusted (moving the weight thresholds) for each particular situation, in order to reduce the rejection rate or improve the classification accuracy. In addition, it is also important to comment that the performance of this method greatly depends on the corpus size. It requires of a large train corpora to calculate confident weights for the terms. For this reason, one of the main directions for future work is the application of semi-supervised learning strategies.

Acknowledgements. This work was done under partial support of CONACYT project grant 61335.

References

1. Lee, C.-H., Carpenter, B., Chou, W., Chu-Carroll, J., Reichl, W., Saad, A., Zhou, Q.: On Natural Languge Call Routing. Speech Communication 31(4), 309–320 (2000)
2. Carpenter, B., Chu-Carroll, J.: Natural Language Call Routing: A Robust, Self-organizing Approach. In: Proc. ICSLP 1998 (1998)
3. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
4. Gorin, A.L., Riccardi, G., Wright, J.H.: How may I help you? Speech Communication 23, 113–127 (1997)
5. Huang, Q., Cox, S.J.: Automatic Call Routing with Multiple Language Models. In: Proc. Workshop on Spoken Language Understanding for Conversational Systems, Boston (May 2004)
6. Kuo, H.-K., Lee, J., Zitouni, C.-H., Fosler-Lussier, I.E.: Discriminative training for call classification and routing. In: ICSLP 2002, pp. 1145–1148 (2002)
7. Garfield, S., Wermter, S., Devlin, S.: Spoken Language Classification using Hybrid Classifier Combination. International Journal of Hybrid Intelligent Systems 2(1), 13–33 (2005)
8. Tang, M., Pellom, B., Hacioglu, K.: Call-Type Classification And Unsupervised Training For The Call Center Domain. In: Automatic Speech Recognition and Understanding IEEE Workshop, pp. 204–208. ASRU 2003 (2003), ISBN 0-7803-7980-2
9. Haffner, P., Tur, G., Wright, J.H.: Optimizing SVMs for complex call classification. In: International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2003, vol. 1, pp. 632–635. IEEE, Los Alamitos (2003)
10. Schapire, R.E., Singer, Y.: BoosTexter: A Boosting-based System for Text Categorization. Machine Learning 39, 135–168 (2000)

Spanish Temporal Expressions: Some Forms Reinforced by an Adverb*

Sofía N. Galicia-Haro

Faculty of Sciences, UNAM, University City, Mexico City, Mexico
sngh@fciencias.unam.mx

Abstract. We consider temporal expressions formed by a noun of time reinforced by an adverb of time, in order to extend the range of annotation of temporal expressions. Fro this, we analyze some groups to build finite state automata to automatically determine such type of expressions. To determine the extent of annotation, we present a method for obtaining the most preferable variants from Internet. We analyze some groups to capture their meaning.

1 Introduction

Different forms in which time is expressed by natural languages raise interesting challenges for its representation as knowledge and for reasoning in general. But before trying to solve these problems, it is necessary to solve the initial problem of automatic recognition of time references in natural language. However, description of these expressions is still far from being complete, since there is a great productivity of temporal expressions in natural languages.

Automatic processing of temporal expressions has to cope with several problems: they are difficult to recognize because of their great variety, their variants could make harder to compare them, their meaning could be ambiguous or difficult to determine. However, this processing is required for many applications, such as machine translation, information extraction, question answering, etc. [2].

In the field of computational linguistics, some practical advances have been achieved by means of the compilation of annotated text collections, to develop and to prove methods. In this work, we are interested in the annotation of temporal expressions. Although diverse investigators have devoted their efforts in developing standards for the annotation of temporal expressions in English, Italian and other languages, the structural variety of these expressions makes us conclude that a great work is still missing to cover all possible temporal expressions in a language, considering all the varieties of temporal information expressed in human communication.

A well-known example of analysis of temporal expression annotation is [5]. Its authors presented guidelines intended to support a variety of applications in the performance of some useful tasks. As authors pointed out, the guideline was not intended to represent all the varieties of temporal information conveyed in natural language communication. The authors were interested in temporal expressions that reference calendar dates, times of day, or durations. They considered lexical triggers to identify

* Work partially supported by Mexican Government (CONACyT and SNI).

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 193–203, 2008.

the temporal expressions. Lexical trigger is a word or numeric expression whose meaning conveys a temporal unit or concept. To be a trigger, the referent must be able to be oriented on a timeline, or at least oriented with relation to a time (past, present, future).

In this work, we are interested in analyzing other different temporal expressions that no fulfill the previous trigger characteristics. These phrases are recognized with an initial adverb, for example: *around*, *still*, and they end with a noun of time as *year*, *month*, for example: *aún en los últimos años* 'still in the last years', *aún en los setentas* 'still in the seventies', *alrededor de año medio* 'about year and a half', *alrededor de hace 20 años* 'about twenty years ago'

The analysis of this type of expressions was motivated by our interest in extending the range of annotation of temporal expressions. We found that this type of phrases present interesting cases, there is for example a group that describe person's age. Since annotation involves determining the kind of expressions to annotate we analyzed some groups to build their corresponding finite state automata to automatically determine them. The deterministic finite automata of person's age should be also useful in question answering and machine translation tasks among others.

For a more detailed annotation we need to capture the meaning of the temporal expressions. We found that some adverbs correspond to fuzzy quantifiers, for example *alrededor* (about) and the interpretation of the phrases where they appear require an approximate definition of the quantifier since a natural language system should be able to anchor each event to a particular time, as well as order the events with respect to each other.

In Section 2 we describe the structure of the phrases we are interested in and how we obtained the initial materials for the analysis carried out in this paper. Then in Section 3 to determine the extent of annotation we describe the method to obtain variants of such phrases from Internet. Section 4 presents our preliminary results: person age group and the interpretation of a fuzzy quantifier.

2 Adverbs and Temporal Expressions

Adverbs of time (for example: *before, today, after, tomorrow*) create cohesion and coherence by forming time continuity in the events expressed in texts [11]. They are closely associated with narrative texts but they also appear in newspaper texts, for example: *Senators will approve today in the morning a schedule for the debate, after the plenary session the coordinators will meet to establish the regulation and tomorrow the tribune will be liberated.*

Researchers have been interested in adverbs of time included in temporal expressions, for example [5] considered some adverbs as: *monthly*, *annual*, *semiannual*. For these temporal adverbs and frequency expressions like: *per year*, *a year*, they found that it is often possible to discern the timeframe in which these sets occur, based on the document's context. Their work was dedicated to annotation and the full extent of the phrases also included all post modifiers of the time expression, among which are adverbs, prepositional phrases, and dependent clauses. Authors considered the adverbs: *currently, lately, hourly, daily, monthly, ago* as Triggers and the adverbs: *earlier, immediately, instantly, forthwith, meanwhile*, as Non-triggers. Markable

expressions are the expressions that should be annotated. To be markable, the syntactic head of the expression must be an appropriate lexical trigger. Each lexical trigger is a word or numeric expression whose meaning conveys a temporal unit or concept, such as "day" or "monthly." Furthermore, to be a trigger, the referent must be able to be oriented on a timeline, or at least oriented with relation to a time (past, present,future).

In [1] authors report a research on temporal adverbs, limited to those formed by a preposition, a determiner and the TimeNoun, for example *a lo largo de* 'through', in addition they considered modifiers to such phrases. They processed a family of Portuguese and Spanish compound temporal adverbs, aiming at building finite state transducers to translate them from one language into the other. Authors had to cope with the large number of combinations involved and their complexity, since it is not easy to list them in full.

Works such as [1, 5, 13] have considered temporal expressions, though not of the type we are interested in. We consider temporal expressions related to Spanish adverbs of time that mostly do not correspond to triggers to extend the range of temporal expressions annotation. We chose to analyze the Spanish temporal phrases that initiate with an adverb of time (AdvT) and end with a noun of time (TimeN), for example:

aún a principios de los años 90 'still at the beginning of the 90s'
aún en nuestros días 'still in our days'
aún hoy día, aún hoy en día 'still nowadays'
alrededor de los 20 años 'about 20 years'
alrededor de los años 20 'about the 20s'

These phrases present interesting issues, we could observe the relation between the groups of words in the following examples:

1) *A sus 80 años Juan tiene pulso de cirujano*
2) *Aún a sus 80 años Juan tiene pulso de cirujano*
3) *Hoy a sus 80 años Juan tiene pulso de cirujano*

The sentences describe the same main fact: *John, who is 80 years old, has surgeon's pulse*, but they tell us something else when we introduce a modifier (*aún* 'still', *hoy* 'today') in each one: they argue for different conclusions.

2) *Still at 80 years old, John has surgeon's pulse* ⇒ in spite of his age he can have it
3) *Today at 80 years old, John has surgeon's pulse* ⇒ today he has it

The adverbs still and today make such conclusions obligatory and reinforce the meaning of time in different forms.

Resources. For our analysis, we selected the texts of four Mexican newspapers that are daily published in the WEB with a considerable part of their complete publication. The texts correspond to diverse sections: economy, politics, culture, sport, etc. from 1998 to 2002. The text collection has approximately 60 million words [6].

We developed a program to extract the sentences matching the following pattern:

AdvT–something–TimeN

where:

something – corresponds to a list of up to five words[1]
TimeN – corresponds to the following nouns of time: *año* 'year', *mes* 'month', *día* 'day', *hora* 'hour', *minuto* 'minute', *segundo* 'second'
AdvT – corresponds to adverbs of time, obtained from [12][2]:

actualmente, adelante, ahora, alrededor, anoche, antaño, anteayer, anteriormente, antes, aún, ayer, constantemente, cuando, de antemano, dentro, dentro de poco, después, en seguida, enseguida, entonces, finalmente, hasta este momento, hogaño, hoy, inmediatamente, instantáneamente, jamás, luego, mañana, más tarde, más temprano, mientras, mientras tanto, momentáneamente, nunca, ocasionalmente, por horas, posteriormente, previamente, prontamente, pronto, recién, recientemente, siempre, simultáneamente, tarde, temprano, todavía, últimamente, una vez, ya

We extracted all sentences accomplishing the AdvT–something–TimeN pattern and the TimeN–something–AdvT pattern since there could be phrases where the style changes the word order, for example: *año nueve meses después* (lit. year nine months later). We found 83319 sentences with these characteristics from 3418375 sentences.

3 Obtaining More Variants

From the 83319 sentences, we manually get approximately five examples for the more numerous cases; i.e. for the cases where the AdvT was: *ahora* 'now', *antes* 'before', *aún* 'still', *todavía* 'still', *actualmente* 'nowadays', *alrededor* 'about'. With these temporal phrases we searched in Internet for more examples [4].

We would want examples similar to those obtained from newspaper texts but with more variety to build finite state automata (FSA) that could automatically determine these types of temporal phrases. To obtain such word sequences we apply the method described in Figure 1, using the Google search engine tool limited to Spanish language and the asterisk feature [3]. For example: for the phrase *aún en su primer año* 'still in his first year' the method in step 1) search for "*aún* en * *año*" in the first ten pages, obtaining the following new phrases:

aún en aquellos años, aún en esos años, aún en muchos años, etc.
aún en siete años, aún en cien años, etc.

where some new determinants, adjectives, and numbers were obtained. In addition some phrases not corresponding to one temporal phrase were picked up, for example: *aún en aquellos ambientes y años* 'still in those ambiences and years', this type of phrases are eliminated in the manual identification at the end of the total process.

In step 2 the phrases are classified and in step 3 one example of the classes obtained is the string "*aún en aquellos * años"*. In step 3.2 with a new process more adjectives like: *primeros, tiernos, oscuros, últimos* and more numbers are collected.

[1] From the collection, there is no relation between the AdvT and the TimeN when a larger quantity of words appears between them.
[2] Adverbs related to specific times were neglected, for example: biannually, monthly, daily, etc.

```
SEARCH(C)
For each phrase of type ADV-*-NounT or string-*-NounT in C
       1) Obtain 100 examples from Internet
            1.1) D = {examples excepting such where * in-
                 cludes verbs or punctuation}
            1.2) Print D
       2) Classify them according such words retrieved by *
       3) For each group of phrases sharing words retrieved
          by *, assign a class Di
            3.1) F = class Di
            3.2) SEARCH(F)
UNTIL no new elements are obtained
```

Fig. 1. Algorithm to obtain variants of temporal expressions

The method considers keeping the words of the right and left contexts, i.e. the previous word of the phrase and a small group of words following the phrase, for example:

ni *aún en el año* ***1914***
Aunque *aún en el año 2007* ***la mayoría***

The process recurred several times until no new repeated phrases are obtained, determining the sequences of words that appear with higher frequency in order to compile more variants of the initial phrase. After this process we select manually the temporal phrases to construct the corresponding FSA.

3.1 Finite State Automata

Temporal expressions determined with the method previously described constitute a set of linguistic expressions that seems particularly well suited for representation by means of FSA [7, 214]; we supposed that both for their modularity and for their relative independence from the sentence in which they appear. For example, the FSA for the phrase *aún* – `something` – *año* is showed in Figure 2, where the blocks names in uppercase letters correspond to groups of specific words. For example, ADJ1 comprises: *varios*, *algunos*, *largos*, *muchos*, etc.

We evaluate the FSAs applying them to the collection of newspaper texts to measure its performance. Results vary depending on the AdvT – something – NounT pattern. Some results show very high recall and success rate. The results for the example shown in Figure 2 were obtained with 38 examples manually annotated, where 31 were correctly obtained by the FSA with a recall of 81.5 From the errors we discovered that left context (*más*, *menos*) changes the meaning of the phrase since *más aún* 'moreover' and *menos aún* 'even less', separate the syntactic relation among the words of the phrase.

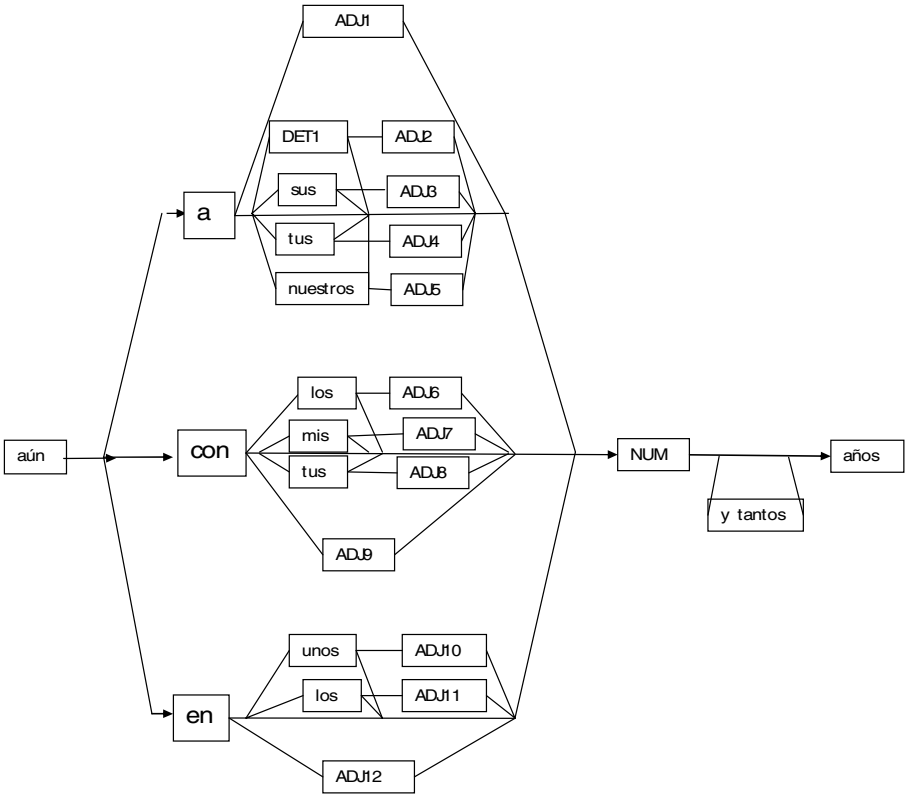


Fig. 2. An example of FSA for the phrase *aún* – something – *año*

Future work will consider the evaluation of over generation of incorrect phrases in the FSA's, although over generation could not be a real problem in annotation because of good rate of text correctness, linguistic adequacy could be reduced and really required for other applications.

Our purpose here, however, is not only to build a set of lexical FSA to be used in recognition of temporal expressions formed with an adverb of time in texts, but also to analyze them as a starting point to interpret them.

When we analyzed these phrases we identified some interesting groups: specific use and adverbs corresponding to a not definite quantifier.

3.2 Phrases Denoting Age of Persons

Usually the age of persons is described by temporal expressions including the time nouns *años* 'years', *meses* 'months' (babies) and they could be extracted in different ways: 1) with a string *de edad* ('old' in English) after the word years, 2) `after the person's name, delimited by commas, 3)` with the strings: *la edad, de edad* 'age' before the quantity of years.

However, there are other temporal expressions that describe the age of persons, for example: *aún a sus 65 años*, 'still at his 65 years', *de alrededor de 20 años*, lit. 'of about 20 year'. We found that among these phrases, some preserves their meaning independent of the context and others require some type of words in context to define the age of a person.

These temporal phrases denote a point in the time line of a person; it could be a point in the time line of the events related in the sentence or a point in a tangential time line.

The following phrases correspond to phrases indicating persons' age that do not depend on context:

aún [*hoy*] *a sus NUM años*
aún a tus NUM años
ahora a mis NUM *años*
aún [*hoy*] *con mis* [*pocos, cortos*] *NUM años*
aún con {*tus, sus*} [*escasos, casi*] *NUM años*
aún con sus NUM años
, *ahora de NUM años***,** (delimited by comma, parenthesis or using a final point)

where NUM represent numbers in words or numerals, [ ] contain optional words and { } indicate variants.

We manually analyzed 100 examples in Internet for every phrase (with an asterisk instead of NUM) and for this group of phrases only two snippets were examples not corresponding to age of a person. In one snippet the phrase was referring to the age of a horse and in the other snippet the phrase was referring to time of experience.

In all these phrases *aún* induces the conclusion: 'in spite of' related to the quantity of years.

The following phrases correspond to phrases indicating persons' age that depend on context:

de alrededor de NUM años
actualmente con NUM años
actualmente de [*unos*] *NUM años*
ahora con [*casi, más de*] *NUM años*
ahora a [*los*] *NUM años*
ahora de NUM meses
alrededor de [*los*] *NUM años*

We also manually analyzed 100 examples in Internet for every phrase and we found that a preceding context of person noun was used in the first phrase to indicate age of a person. The rest of the phrases require a rear context like *de edad* 'of age', *encima* 'above', *a cuestas* 'on the back', *en sus espaldas* 'in his backs', *a las espaldas* 'to the backs', *a sus espaldas* 'to his backs', to denote age of a person.

We also found quasi fixed expressions like: *actualmente cuenta con NUM años de edad* 'at present it is provided with NUM years of age' referring to the current age of a person.

3.3 Fuzzy Quantifier

We have a good sense of how long a task that we perform frequently should take, but sometimes it is difficult to understand time periods that do not correspond to our daily personal experience. The problem is that we do not know if it is a long time or a short time in a particular context.

Another group of the temporal expressions analyzed corresponds to expressions which denote concepts that cannot be explicitly anchored on the timeline since the adverb has an interpretation of not definite quantifier. For example, *alrededor* 'about' in the phrases *alrededor de 20 años* 'about 20 years', *alrededor de año y medio* 'about year and a half', and for this type of phrases we analyze the possible ways of assigning an approximate quantification.

Zadeh [16] wrote that manipulation of perceptions plays a key role in human recognition, decision and execution processes. His methodology: Computing with words (CW), provides a foundation for a computational theory of perceptions. He points out that a basic difference between perceptions and measurements is that, in general, measurements are crisp whereas perceptions are fuzzy.

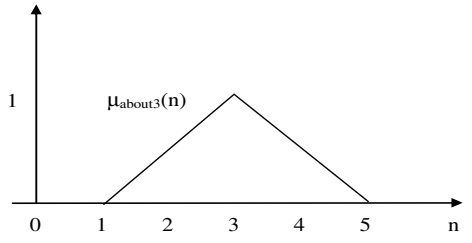
In his Computational theory of perceptions, words play the role of labels of perceptions and, more generally, perceptions are expressed as propositions in a natural language. CW-based techniques are employed to translate propositions expressed in a natural language into what is called the Generalized Constraint Language. In this language, the meaning of a proposition is expressed as a generalized constraint, X isr R, where isr, is a variable copula which defines the way in which R constrains X.

More specifically, the role of R in relation to X is defined by the value of the discrete variable r. When r takes the value d, the constraint is *disjunctive* (possibilistic) and isd abbreviated to is: X is R, in which R is a fuzzy relation which constrains X by playing the role of the possibility distribution of X. More specifically, if X takes values in a universe of discourse, U = {u}, then Poss{X = u} = μR(u), where μR is the membership function of R, and ΠX is the possibility distribution of X, that is, the fuzzy set of its possible values [15]. In schematic form:

$$\text{X is R} \begin{cases} \Pi x = \text{R} \\ \text{Poss}\{\text{X} = \text{u}\} = \mu_{\text{R}}(\text{u}) \end{cases}$$

Fuzzy logic provides a foundation for the development of new tools for dealing with natural languages and knowledge representation. What is used most frequently in practical applications of fuzzy logic is the *basic interpolative rule*, which is a special case of the compositional rule of inference applied to a function which is defined by a fuzzy graph. A fuzzy graph may be viewed as a disjunction of cartesian granules. In CW, a granule, g, which is the denotation of a word, w, is viewed as a fuzzy constraint on a variable. In essence, a fuzzy graph serves as an approximation to a function or a relation [15].

Now let us turn to which fuzzy quantifier corresponds to the given natural language quantifier *around* 'about'. In [10] authors present a simple representation of fuzzy quantifiers given in Figure 3.



X	1	2	3	4
$\mu_{about3}(X)$	0	0.6	1.0	0.6

Fig. 3. A representation for the quantifier *about 3*

In Figure 3 we could note that the values in the set—1, 2, 3, 4, 5—have various degrees of membership in the set of "*about 3*". For instance, 1 and 5 have the least degree of membership and 3 has the greatest degree of membership. The degrees of membership could be represented as shown in the table of Figure 3.

Fuzzy graphs not only allow easy interpretation but in addition finding regions of interest. The triangular membership function is the most frequently used function and the most practical, but other shapes are also used. One is the trapezoid, as shown in Figure 4

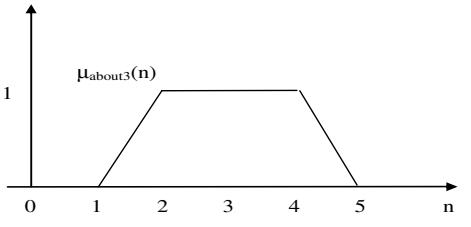


Fig. 4. A representation for the quantifier *about 3*

Trapezoid seems a better representation since the meanings of the adverb *alrededor* in temporal phrases (approximately; making a detour, concerning something; little earlier or after) denote a region with the same degree of membership [8, 9].

To analyze some examples we consider the following hierarchy of time units: years, months, weeks, days, hours, minutes, and seconds. We chose the following examples to analyze the region and to propose some heuristics for the trapezoid width:

- The reference is not a sharp number, for example: *alrededor de* NUM *años*, where NUM is several tens, hundreds, thousands, etc.
  Since decimal system is the most widely used numeral system, one very common fraction is a ten percent. If we think in not sharp quantities, they often are suspicious of being estimations done without the suitable information so we could assign a ten percent around the quantity as a minimum estimation.

- The reference is a sharp number, for example: *alrededor de* NUM *años*, where NUM is a sharp number as 11, 377, 1968, etc.

  When we think in sharp numbers it is assumed that they show more veracity or definition. We could not think in a ten percent for the region around NUM in this

case, for example: 196 more or less for 1968 years. We should then focus the reference in year, a region around one year as a minimum. For the phrase *alrededor de una hora y 15 minutos*, we should then focus on the smaller unit, i.e. minutes for the reference.

If we consider months, weeks, days, hours, minutes and seconds instead of years, we also could apply the previous heuristics.

- The reference is a point in time, for example: *alrededor del medio día*, *alrededor de las doce del día*.

The phrases include a clear point in time: midday, noon. We should then focus in the specific time represented by these special words. Even if it is mentioned something like 'a day' the reference should be minutes as a minimum, since midday defines the 12 a.m. The heuristic now could be based on the time noun considered as a sharp number. But it requires the compilation of relations between this type of words and time units.

- The reference is one unit, for example: *alrededor de un año y medio, alrededor del mes de marzo*.

The phrases express a unit: year, month, etc. We should then focus in a reference of a lower hierarchically common unit of time than that mentioned. Since year is mentioned, month should be considered for the region.

We analyzed these heuristics on a small collection of phrases and the results were acceptable but it is necessary a wide phrase set and a more detailed analysis of results to consider that they could be applied to any natural language application.

Some examples where the reference is another fuzzy quantifier were not considered yet, for example, the indefinite determinants in: *alrededor de unos minutos*, *alrededor de unas horas*.

4 Conclusions

The variety in the structure of temporary expressions makes necessary the analysis of different combinations of classes of words. We analyzed temporal expressions including a noun of time that are recognized by an initial adverb of time, to automatically determine such type of expressions and to annotate a corpus.

Several elements that compose these types of phrases present some modularity, and their combinations make them particularly suitable for a formal description by means of finite state automata. We present a method to enrich the variety of each temporal phrase when only few examples are compiled. The method automatically brings more similar examples but since it does not include complex tools, manually identification was performed at the end of the process.

The automatic identification of these phrases and their interpretation will directly benefit natural language processing tasks as: response to questions; relation, characterization and tracking of events; visualization of events in lines of time; generation of phrases; translation, etc. We found a group that could be interpreted as person's age and another group where adverb could be described as a fuzzy quantifier. For the last group some heuristics for evaluating quantifying expressions have been presented. For the first group we analyzed the context of this type of temporal expressions to

identify the phrases where its meaning is context independent and to define the specific context for other expressions.

Since this kind of expressions present interesting cases, future work should consider the analysis of other groups in the same direction: capturing their meaning and determining the extent of the phrases.

References

1. Baptista, J., Dolors, C.: Compound Temporal Adverbs in Portuguese and in Spanish. In: Ranchhod, E., Mamede, N.J. (eds.) PorTAL 2002. LNCS (LNAI), vol. 2389, pp. 133–136. Springer, Heidelberg (2002)
2. Bolshakov, I.A., Gelbukh, A.: Computational Linguistics: Models, Resources, Applications. IPN–UNAM–FCE, 186 p (2004)
3. Gelbukh, A., Bolshakov, I.A.: Internet, a true friend of translator: the Google wildcard operator. International Journal of Translation 18(1–2), 41–48 (2006)
4. Gelbukh, A., Sidorov, G., Chanona-Hernández, L.: Compilation of a Spanish representative corpus. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 285–288. Springer, Heidelberg (2002)
5. Ferro, L., Gerber, L., Mani, I., Sundheim, B., Wilson, G.: TIDES 2003 Standard for the Annotation of Temporal Expressions. MITRE Corporation (April 2004)
6. Galicia-Haro, S.N.: Using Electronic Texts for an Annotated Corpus Building. In: 4th Mexican International Conference on Computer Science, ENC 2003, Mexico, pp. 26–33 (2003)
7. Gross, M.: The Construction of Local Grammars. In: Schabes, Y., Roche, E. (eds.) Finite State Language Processing, pp. 329–354. MIT Press/Bradford, Cambridge/London (1997)
8. Knoll, A., Glöckner, I.: Fuzzy quantifiers: A linguistic technique for data fusion. In: Proceedings of ISSEK Workshop (2000)
9. Glöckner, I., Knoll, A.: A formal theory of fuzzy natural language quantification and its role in granular computing. In: Pedrycz, W. (ed.) Granular Computing: An Emerging Paradigm. Physica-Verlag (2001)
10. Liétard, L., Rocacher, D.: Evaluation of Quantified Statements using Gradual Numbers. In: Galindo, J. (ed.)Handbook of Research on Fuzzy Information Processing in Databases, Information Science Reference (2008)
11. Llido, D., Berlanga, R., Aramburu, M.J.: Extracting temporal references to assign document event-time periods. In: Mayr, H.C., Lazanský, J., Quirchmayr, G., Vogel, P. (eds.) DEXA 2001. LNCS, vol. 2113, pp. 62–71. Springer, Heidelberg (2001)
12. Real Academia Española: Diccionario de la Real Academia Española, 21 edición (CD-ROM), Espasa, Calpe (1995)
13. Saquete, E., Martinez-Barco, P.: Grammar specification for the recognition of temporal expressions. In: Proceedings of Machine Translation and multilingual applications in the new millennium, MT 2000, Exeter, UK, pp. 21.1–21.7 (2000)
14. Schabes, Y., Roche, E. (eds.): Finite State Language Processing. MIT Press/Bradford, Cambridge/London (1997)
15. Zadeh, L.A.: Fuzzy logic and the calculi of fuzzy rules and fuzzy graphs: A precis. — Multiple Valued Logic, vol. 1, pp. 1–38. Gordon and Breach Science Publishers (1996)
16. Zadeh, L.A.: From Computing with Numbers to Computing with Words—from Manipulation of Measurements to Manipulation of Perceptions. International Journal of Applied Math. and Computer Science 12(3), 307–324 (2002)

A Soundex-Based Approach for Spoken Document Retrieval

M. Alejandro Reyes-Barragán, Luis Villaseñor-Pineda, and Manuel Montes-y-Gómez

Laboratorio de Tecnologías del Lenguaje,
Instituto Nacional de Astrofísica, Óptica y Electrónica, México
{alejandroreyes,villasen,mmontesg}@inaoep.mx

Abstract. Current storage and processing facilities have caused the emergence of many multimedia repositories and, consequently, they have also triggered the necessity of new approaches for information retrieval. In particular, spoken document retrieval is a very complex task since existing speech recognition systems tend to generate several transcription errors (such as word substitutions, insertions and deletions). In order to deal with these errors, this paper proposes an enriched document representation based on a phonetic codification of the automatic transcriptions. This representation aims to reduce the impact of the transcription errors by representing words with similar pronunciations through the same phonetic code. Experimental results on the CL-SR corpus from the CLEF 2007 (which includes 33 test topics and 8,104 English interviews) are encouraging; our method achieved a mean average precision of 0.0795, outperforming all except one of the evaluated systems at this forum.

1 Introduction

Nowadays, thanks to the falling price of storage media and the rising capacities for information generation, the amount of available multimedia repositories is increasing. In such a situation, it is clear that new retrieval methods are required to search through all this information.

In particular, this paper focuses on the task of *spoken document retrieval* (SDR), which consists of searching relevant information for general user queries from a collection of automatic transcriptions of speech. In this case, the transcriptions may come from many different kinds of audio sources, such as radio and television news programs, political discourses, scientific conferences, business meetings, and interviews.

The conventional approach for SDR integrates speech recognition and information retrieval technologies. It first applies an automatic speech recognition (ASR) process to generate text transcriptions from speech (i.e., the spoken documents), and then it makes use of traditional –textual– information retrieval (IR) techniques to search for the desired information.

The main drawback of this approach is that it greatly depends on the quality of the generated transcriptions, which are far from being perfect. Current ASR methods tend to generate several transcription errors (such as word substitutions, insertions and deletions), producing word error rates that vary from 20% to 40% in accordance to the

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 204–211, 2008.

kind of discourse. Particularly, most of these errors are related to the treatment of out-of-vocabulary words, which in the majority of cases are substituted for a phonetically similar –known– word.

Surprisingly, in spite of the frequent transcription errors, the conventional approach for SDR has achieved more or less satisfactory results. As Allan described in [1], it seems that the impact of the transcription errors is minimal when the queries and/or spoken documents are sufficiently large. Nevertheless, for other kind of scenarios (for instance, those considering short queries and/or spoken documents), these errors have a strong influence on the retrieval results. In these cases, the substitution or elimination of some words may really alter the retrieval process. For example, if the speech utterance "Unix Sun Workstation" is incorrectly transcribed by the ASR process as "unique set some workstation", then it would be impossible to retrieve this phrase by the query "Unix".

Most recent works on SDR have paid no attention to the transcription errors and have mainly concentrated on evaluating the usefulness of IR techniques such as query expansion, relevance feedback and information fusion [2, 3, 4, 5]. In contrast with these works, in this paper we propose a new document representation based on a *phonetic codification* of the automatic transcriptions. The purpose of this representation is to reduce the impact of the transcription errors by characterizing words with similar pronunciations through the same phonetic code. In particular, we propose using the *Soundex codes* [6] to enrich the representation of transcriptions. This way, the example transcription phrase "unique set some workstation" will be also represented by the codes U52000 S30000 S50000 W62300, allowing its retrieval by phonetically similar words such as Unix (with code U52000) or sun (with code S50000).

It is important to mention that the idea of using a phonetic codification for indexing automatic transcriptions of speech is not completely new. There was some attempt to use the Soundex codes for indexing names with the aim of finding all their pronunciation variants [7]. Continuing this idea, in this paper we extend this early approach by applying the phonetic codification indiscriminately to all transcription words. To our knowledge, this is the first time that the Soundex codification has been used and evaluated in the task of SDR.

The rest of the paper is organized as follows. Section 2 introduces the task of SDR and presents the evaluation results from the Cross-Language Speech Retrieval Track of the 2007 Cross-Language Evaluation Forum. Section 3 describes the proposed enriched representation for transcriptions as well as the Soundex codification algorithm. Section 4 presents the evaluation results. Finally, Section 5 gives our conclusions and describes some ideas for future work.

2 Spoken Document Retrieval

The task of SDR consists of searching relevant information for general user queries from a collection of automatic transcriptions of speech. Research in this task has been mainly fostered by two international evaluation conferences, initially by the TREC[1] [8, 9] and more recently by the CLEF[2] [10, 11]. The first one used news recordings, whereas the second considers a collection of spontaneous conversational speeches.

[1] The Text REtrieval Conference, http://trec.nist.gov.
[2] The Cross Language Evaluation Forum, http://www.clef-campaign.org.

In this paper, we use the data from the Cross-Language Speech Retrieval Track of CLEF 2007 (hereafter referred as CL-SR 2007). This corpus included 8,104 transcriptions of English interviews as well as 96 query topics, 63 for tuning and validation, and 33 for evaluation.

It is important to mention that for each interview there were provided three different automatic transcriptions, each of them having a different word error rate (WER): ASR03 with a WER of 44%, ASR04 with a WER of 38%, and ASR06 with a WER of 25%. In addition, each interview transcription was annotated using two different sets of automatically extracted keywords (AK1 and AK2).

Table 1 shows the evaluation results from this track (more details can be consulted in [11]). From these results, it is clear that SDR is a very complex task; the MAP (mean average precision calculated at the first 100 documents) scores are very low, much lower than those from traditional textual IR (which are around 0.4).

Table 1. Evaluation results from the English monolingual task of CL-SR 2007

Participating Team	*Used Information*	*MAP*
University of Ottawa [3]	AK1,AK2,ASR04	0.0855
Dublin City University [2]	AK1,AK2,ASR06	0.0787
Brown University [4]	AK1,AK2,ASR06	0.0785
University of Chicago [5]	AK1,AK2,ASR06	0.0571
University of Amsterdam [12]	AK2,ASR06	0.0444

The complexity of SDR is due to several factors. First, the automatic transcriptions are not perfect, that is, they contain many errors such as word substitutions, insertions and deletions. Second, the used vocabulary tends to be smaller than that from written documents. Third, the indexing units are commonly speech segments rather than complete documents. Regardless these additional complexities, the participating teams at the CL-SR 2007 mainly focused on applying traditional IR techniques. They used different weighting schemes [3], several query expansion and relevance feedback techniques [2, 3, 4], as well as some information fusion approaches [3]. In other words, none of them did something special for tackling the transcription errors. Moreover, different to our proposal, none of them took into consideration any kind of phonetic information from transcriptions.

3 An Enriched Representation for Spoken Documents

As we previously mentioned, the proposed approach for SDR relies on a *phonetically enriched representation* of the automatic transcriptions (spoken documents). This representation aims to reduce the impact of the transcription errors by characterizing words with similar pronunciations through the same phonetic code.

The construction of the enriched representations considers the following actions:

1. Compute the phonetic codification for each transcription using the Soundex algorithm (refer to Section 3.1).
2. Combine transcriptions and their phonetic codifications in order to form the enriched document representations. This way, each document is represented by a mixed bag of words and phonetic codes.

3. Remove unimportant tokens from the new document representations. In this case, we eliminate a common list of stopwords as well as the most frequent phonetic codes (refer to Section 4).

In order to clarify this procedure, Table 2 illustrates the construction of the enriched representation for the transcription segment "*…just your early discussions was roll wallenberg's uh any recollection of of uh where he came from and so…*", which belong to the spoken document with id=VHF31914-137755.013 from the CL-SR 2007 corpus.

Table 2. Example of an enriched document representation

Automatic transcription	…just your early discussions was roll wallenberg uh any recollection of of uh where he came from…
Phonetic codification	... J23000 Y60000 E64000 D22520 W20000 R40000 W45162 U00000 A50000 R24235 O10000 O10000 U00000 W60000 H00000 C50000 F65000 ...
Enriched representation	{just, early, discussions, roll, wallenberg, recollection, came, E64000, D22520, R40000, W45162, R24235}

It is important to mention that, in order to take advantage of the proposed representation in the retrieval process, it is also necessary to construct the enriched representation of queries. For that purpose we apply the same procedure than the one used for constructing the enriched representations of transcriptions. Table 3 shows the enriched representation of a query from the CL-SR 2007 corpus.

Table 3. Example of an enriched query representation

Original query	Eyewitness accounts that describe the personalities and actions of Raoul Wallenberg and Adolf Eichmann
Phonetic codification	E35200 A25320 T30000 D26100 T00000 P62543 A53000 A23520 O10000 R40000 W45162 A53000 A34100 E25500
Enriched representation	{eyewitness, accounts, personalities, actions, raoul, wallenberg, adolf, eichmann, E35200, A25320, P62543, A23520, R40000, W45162, A34100, E25500}

From this example, it is interesting to notice that the usage of the phonetic codifications allows improving the matching between transcription and query, since the words "roll" and "Raoul" were both represented by the same phonetic code (R40000). For this particular case, we presume that the word "roll" was obtained from an incorrect transcription of the word "Raoul".

3.1 The Soundex Codification Algorithm

Phonetic codifications attempt to represent words with similar pronunciations by the same code. Among all existing phonetic codification algorithms, *Soundex* is the most widely known. It was originally proposed for dealing with the problem of having different spelling variations of the same name (e.g., Lewinsky vs. Lewinskey) [6], and since then it has been applied in several database applications for indexing surnames, for instance, it has been used in the U.S. census.

The Soundex algorithm is based on the phonetic classification of human speech sounds (bilabial, labiodental, dental, alveolar, velar, and glottal), which in turn are based on where we put our lips and tongue to make sounds. The algorithm itself is straightforward since it does not require of backtracking or multiple passes over the input word. This algorithm is as follows:

1. Capitalize all letters in the word and drop all punctuation marks.
2. Retain the first letter of the word.
3. Change all occurrence of the following letters to '0' (zero): 'A', E', 'I', 'O', 'U', 'H', 'W', 'Y'.
4. Change letters from the following sets into the given digit:
   - 1 = 'B', 'F', 'P', 'V'
   - 2 = 'C', 'G', 'J', 'K', 'Q', 'S', 'X', 'Z'
   - 3 = 'D','T'
   - 4 = 'L'
   - 5 = 'M','N'
   - 6 = 'R'
5. Remove all pairs of equal digits occurring beside each other from the string resulted after step (4).
6. Remove all zeros from the string that results from step (5)
7. Pad the string resulted from step (6) with trailing zeros and return only the first six positions. The output code will be of the form <uppercase letter> <digit> <digit> <digit> <digit> <digit>.

Using this algorithm, both "Robert" and "Rupert" return the same string "R16300", whereas "Rubin" yields "R15000".

4 Experimental Results

4.1 Experimental Setup

This section presents some experiments that allow evaluating the usefulness of the proposed representation. In all these experiments, we used the data from the CL-SR 2007 task [11]. In particular, we considered the automatic transcription ASR06 (with a word error rate of 25%), and the sets of automatic keywords AK1 and AK2. Besides, we made use of the Indri search engine [13], which provided us with the functionalities for term weighting and query expansion[3].

It is also necessary to mention that, in accordance to the proposed representation, in all experiments we eliminated a set of stopwords as well as a set of high frequency phonetic codes. Mainly, we decided to eliminate the same proportion of both items. In this way, we eliminated 319 stopwords[4], which correspond to 75% of the word items, and 265 phonetic codes, which approximately represent the same percentage from the whole set of used phonetic codes.

[3] We applied a blind query expansion approach that incorporated to the original query the ten most frequent terms from the first ten ranked documents.

[4] The list was taken from the IR resources of the Department of Computing Science at the University of Glasgow (www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words).

On the other hand, the evaluation was carried out using the *MAP* (mean average precision) and *precision@10* measures. These measures are calculated as follows:

$$MAP = \frac{1}{|Q|} \sum_{\forall q \in Q} \frac{\sum_{r=1}^{N} (P(r) \times rel(r))}{\text{number of relevant documents}}$$

$$precision@10 = P(r = 10)$$

where Q is the set of test questions, N is the number of retrieved documents, r indicates the rank of a document, *rel*() is a binary function on the relevance of a given rank, and P() is the precision at a given cut-off rank. This way, the precision@10 indicates the percentage of correct items from the first ten retrieved documents.

4.2 Results

The first experiment aimed to determine the *pertinence of the phonetic codification*. In order to do that, we performed the SDR considering only textual information (the transcriptions along with automatic keywords) as well as using the phonetic codification by itself. Table 4 shows the achieved results.

Table 4. Text-based vs. Phonetic-based retrieval

	SRD using only text (ASR06 + AK1 + AK2)	**SDR using only phonetic codes** (phonetic codification of ASR06)
MAP	0.108	0.081
Precision@10	25.4%	19.4%

This experiment showed that the phonetic-based retrieval could identify an important number of relevant documents, even though its results were inferior to those from the text-based approach. In addition, this experiment allowed us observing the complementarity of both approaches: together they retrieved 2240 relevant documents, nevertheless they only shared 1674, indicating that their results are complementary in 27%.

Based on this last observation, the second experiment attempted to evaluate the *effectiveness of the proposed representation*, which combines textual and phonetic information. For this experiment, the weight of words was defined as the double than the weight of phonetic codes. Table 5 shows the results from this experiment.

Table 5. Combining transcriptions and their phonetic codifications

	Proposed representation	**Improvement over Text-based SDR**
MAP	0.116	+ 6.8%
Precision@10	28.1%	+ 10.6%

From these results, it is possible to conclude that the combination of textual and phonetic information stated by the proposed representation improves the SDR

process. Nevertheless, it is necessary to do an extensive analysis in order to determine the kind of transcription errors that were handled by the phonetic codification.

Finally, a third experiment was carried out to compare our proposal against other current state-of-the-art methods evaluated on the CL-SR 2007 test corpus. In this experiment, we use the test corpus consisting of 33 topics. Table 6 shows the results from this experiment. It can be observed that our proposal achieved a MAP of 0.0795, outperforming all except one of the evaluated systems at this forum. However, it is important to comment that our proposal (using a phonetically enriched document representation) could be used in conjunction with any one of these methods, probably leading to a better performance.

Table 6. Comparison of our results against methods from CL-SR 2007

Team	***Used Information***	***MAP***
University of Ottawa	AK1,AK2,ASR04	0.0855
Our proposal	AK1,AK2,ASR06	0.0795
Dublin City University	AK1,AK2,ASR06	0.0787
Brown University	AK1,AK2,ASR06	0.0785
University of Chicago	AK1,AK2,ASR06	0.0571
University of Ámsterdam	AK2,ASR06	0.0444

5 Conclusions

In this paper, we have proposed an *enriched representation for spoken documents* that is specially suited for the IR task. This new representation is based on a phonetic codification of the automatic transcriptions, which aims to reduce the impact of the transcription errors by characterizing words with similar pronunciations through the same phonetic code.

Experimental results on the CL-SR 2007 data set are encouraging; our proposal achieved a MAP of 0.0795, outperforming all except one of the evaluated systems at this forum. In addition, these results demonstrated the usefulness of the phonetic codification as well as its complementarity with the textual information.

It is clear that it is necessary to perform more experiments in order to conclude about the advantages of the proposed representation. In particular, we plan to:

- Consider other phonetic codifications (such as Daitch-Mokotoff, NYSIIS, Phonix, Metaphone, and Double Metaphone) and perform additional experiments for determining the most appropriate one.
- Accomplish a detailed analysis of current results in order to determine the kind of transcription errors that were successfully handled by the phonetic codification. Particularly we are interested in studying the treatment of out-of-vocabulary words.
- Use the proposed representation in conjunction with different SDR methods such as those participating at the CL-SR 2007 task.
- Explore the usage of the proposed representation at character n-gram level. In this way, it is possible to take away the word segmentation imposed by the ASR process, and therefore, it is easier to tackle the problems of word insertions and deletions.

Acknowledgements

This work was done under partial support of CONACYT (project grant 61335 and scholarship 212715). We also like to thank the CLEF organizing committee for the resources provided.

References

1. Allan, J.: Perspectives on Information Retrieval and Speech. In: Coden, A.R., Brown, E.W., Srinivasan, S. (eds.) SIGIR-WS 2001. LNCS, vol. 2273. Springer, Heidelberg (2002)
2. Jones, G., Zhang, K., Lam-Adesina, A.: Dublin City University at CLEF 2007: Cross-Language Speech Retrieval (CL-SR) Experiments. Working Notes of the 8th Workshop of the Cross-Language Evaluation Forum (CLEF 2007), Budapest, Hungary, September 19-21 (2007)
3. Alzghool, M., Inkpek, D.: Model Fusion for the Cross Language Speech Retrieval Task at CLEF 2007. Working Notes of the 8th Workshop of the Cross-Language Evaluation Forum (CLEF 2007), Budapest, Hungary, September 19-21 (2007)
4. Lease, M., Charniak, E.: Brown at CL-SR 2007: Retrieval Conversational Speech in English and Czech. Working Notes of the 8th Workshop of the Cross-Language Evaluation Forum (CLEF 2007), Budapest, Hungary, September 19-21 (2007)
5. Levow, G.: University of Chicago at the CLEF 2007 Cross-Language Speech Retrieval Track. Working Notes of the 8th Workshop of the Cross-Language Evaluation Forum (CLEF 2007), Budapest, Hungary, September 19-21 (2007)
6. Odell, M.K., Russell, R.C.: U.S. Patent Numbers 1261167 (1918) and 1435663 (1922). Washington, D.C.: U.S. Patent Office (1918)
7. Raghavan, H., Allan, J.: Using Soundex Codes for Indexing Names in ASR documents. In: Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at Humal Language Technology Conference and North American chapter of Association of Computa Computational Linguistics, Boston, MA, USA, pp. 22–27 (2004)
8. Voorhees, E., Garofolo, J., Jones, K.: The TREC-6 Spoken Document Retrieval Track. In: Proceedings of the Sixth Text Retrieval Conference (TREC-6), Gaithersburg, Maryland, November 19–21 (1997)
9. Garafolo, J.S., Auzanne, C.G.P., Voorhees, E.: The TREC Spoken Document Retrieval Track: A Success Story. In: Proceedings of the RIAO 2000 Conference: Content-Based Multimedia Information Access, Paris, France (2000)
10. White, R., Oard, D., Jones, G., Soergel, D., Huang, X.: Overview of the CLEF-2005 Cross-Language Speech Retrieval Track. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, pp. 744–759. Springer, Heidelberg (2006)
11. Pecina, P., Hoffmannová, P.: Overview of the CLEF-2007 Cross-Language Speech Retrieval Track. Working Notes of the 8th Workshop of the Cross-Language Evaluation Forum (CLEF 2007), Budapest, Hungary, September 19-21 (2007)
12. Huurnink, B.: The University of Amsterdam at the CLEF Cross Language Speech Retrieval Track 2007. Working Notes of the 8th Workshop of the Cross-Language Evaluation Forum (CLEF 2007), Budapest, Hungary, September 19-21 (2007)
13. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A Language-Model based Search Engine for Complex Queries. In: Proceedings of the International Conference on Intelligence Analysis, McLean, VA, May 2-6 (2005)

Textual Entailment Recognition Based on Structural Isomorphism

Diego Uribe

Instituto Tecnológico de la Laguna
División de Posgrado e Investigación
Blvd. Revolución y Cuauhtémoc, Torreón, Coah., MX
duribe@itlalaguna.edu.mx

Abstract. In this paper we define a measure for textual entailment recognition based on structural isomorphism theory applied to lexical dependency information. We describe the experiments carried out to estimate measure's parameters with logistic regression and SVM. The results obtained show how a model constructed around lexical relationships is a plausible alternative for textual entailment recognition.

1 Introduction

Recognition of textual entailment (RTE) is the task of deciding, given two text fragments denoted by Text (T) and Hypothesis (H), whether the meaning of H can be inferred from the meaning of T. In this paper we experiment with a representation of text pairs based on grammatical dependencies that enable us to identify structural similarities. Basically, our method for textual entailment recognition builds upon structure alignment techniques, that is, we analyzed the RTE problem from a graph similarity measure perspective [1]. More specifically, we focus on tree alignment so our algorithm automatically analyses the dependency trees corresponding to each of the sentences of a given text pair. In this way, the main idea is to maximize structural isomorphism.

In short, the technique involved in the experiments show how the differences in the constituent structure of the textual pairs, as well as negation as fundamental linguistic transformations which may alter the inference process are being considered.

In this paper, we first introduce in section 2 the basic elements that we use in our experiments such as the data set and the pre-processing to which we submitted our data collection. Section 3 describes the feature set to be used in our classification method. The results of the experiments based on logistic regression and SVM techniques are described and discussed in section 4. Finally, we conclude in section 5 with some observations and future work.

2 Basic Elements

In this paper we propose a measure for textual entailment recognition based on a set of features whose vector values are determined by the similarity of the

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 212–219, 2008.

dependencies graphs corresponding to each textual pair. Our approach could appear similar to Pazienza's work on textual entailment [2] but there are two key differences: the set of features and the dependency graph. In fact, we define the set of features in a different way and we do not make use of the extended dependency graph. The basic components of the system which support the experimental method are described next.

2.1 The Data

The data set used in this experiment is the RTE collection which consists of seven subsets compiled by human annotators. These seven subsets correspond to different applications where the annotators selected both positive and negative entailment instances in a roughly one to one proportion (i.e. 50%-50% split).

2.2 Pre-processing

Each T-H example represented by a pair of sentences in the dataset used in our experimentation was submitted to a statistical parser: The Stanford Parser[1]. This statistical parser generates typed dependencies parsers from phrase structure parsers in such a way that the structure of the sentence is represented by dependencies between individual words. Since these dependencies represent grammatical relations which convey information in the way of predicate-argument structures, they are also known as very general semantic relations. In this way, it seems plausible to make use of such representation to look for isomorphism between dependencies trees rather than traditional phrase structure trees.

3 Experimental Method

Our work on textual entailment recognition is based on the idea to maximize structural isomorphism. That is, we are looking for a mapping between T-H examples that allow us to detect similar meaning. Specifically, our algorithm attempts to fully automate dependency tree alignment using an approach inspired by Matsumoto et al. [3]. The algorithm takes as input the set of grammar dependencies corresponding to positive and negative instances of textual entailments from which we extract some features, and then based on these features, to classify a new pair of texts as paraphrasing or not. In our particular case, one set of features describes the paraphrase texts, and for each feature f_i, we have a corresponding weight w_i. The features and their computation are described below.

3.1 Feature Extraction

This section describes how we extract features from the dependency trees corresponding to each T-H example. Since we look for structural isomorphism, coupling between dependency trees is estimated according to the similarity among

[1] http://www-nlp.stanford.edu/software/lex-parser.shtml

the different levels of the trees. In other words, for a particular node t_1 we search for a similar node t_2 in such a way that their corresponding descendants share as many similar nodes as possible. Thus, the set of features is represented as a vector where each element denotes the similarity through the different levels of the trees.

In order to set up the feature vector, our coupling nodes process uses a basic similarity function to estimate the association between a pair of lexical terms w_1, w_2 by making use of WordNet as thesaurus:

$$similarity(w_1, w_2) = \begin{cases} 1 & \text{if } (w_1, w_2) \text{ are synonyms in the thesaurus} \\ 0 & \text{otherwise} \end{cases}$$

The similarity score between two nodes is then computed according to the minimum number of children between the nodes. This means that the similarity of two lexical nodes is based on the arguments of the lexical nodes: the number of related arguments determines the similarity score of two lexical nodes. In this way, let be c_1 and c_2 representing the number of children of nodes t_1 and t_2 respectively. Then,

$$similarityScore(t_1, t_2) = \frac{sumCost}{n}$$

where

$$sumCost = \sum_{i=1}^{c_1} \sum_{j=1}^{c_2} similarity(c_i, c_j)$$

and

$$n = \begin{cases} c_1 & \text{if } c_1 \leq c_2 \\ c_2 & \text{otherwise} \end{cases}$$

Table 1 shows a T-H example represented by a pair of sentences and their corresponding dependency trees. The computation of the feature vector, which has been set up according to the mathematical expressions shown above, is the following:

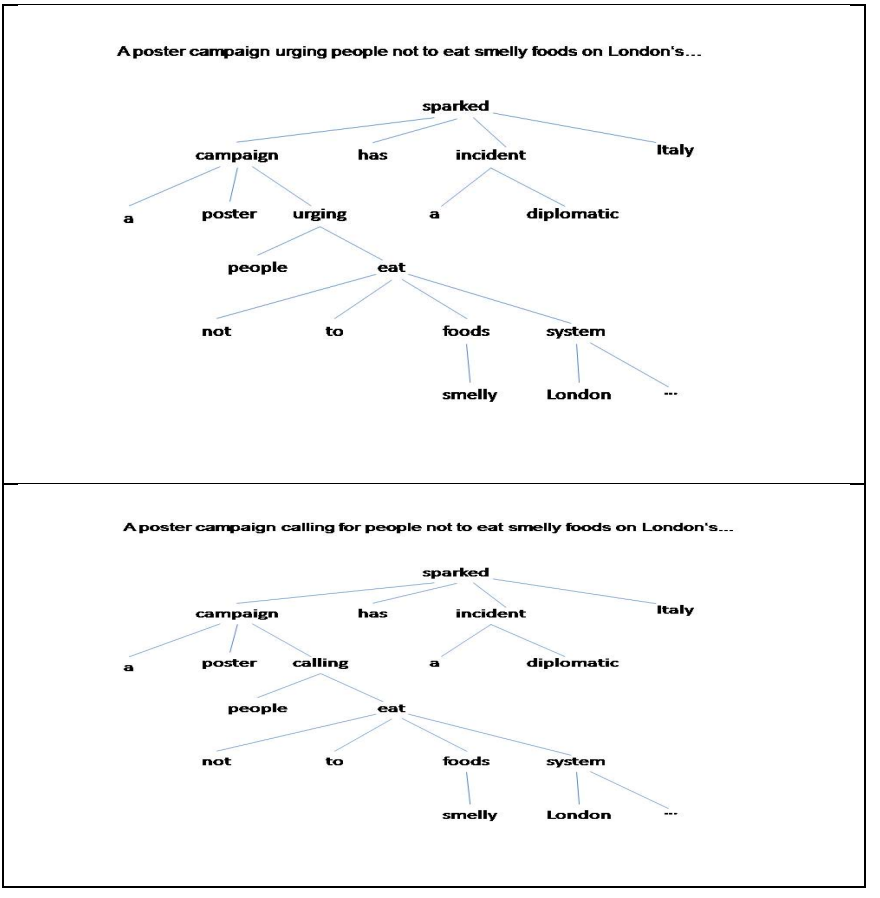
$$Featurevector = [1.0, 1.0, 0.8333, 0.0]$$

As we can see, the feature vector contains four values associated with the four levels of the subtree which, in this particular case, maximizes the structural isomorphism.

3.2 Classification Process

In this section we describe the two steps which constitute our classification process: the search mechanism and the learning method for tagging new pairs of sentences.

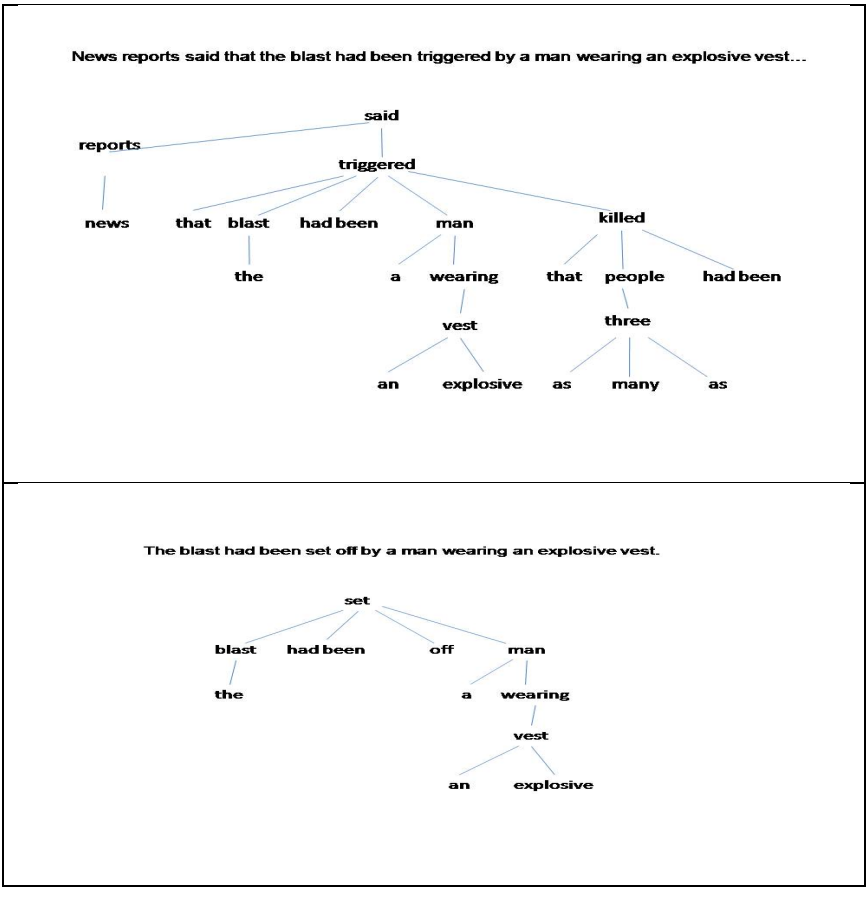
Table 1. Dependency trees for a particular textual pair



Firstly, our algorithm implements top-down, depth-first search. Since there could be more than one coupling between dependency trees, and therefore more than one feature vector for a particular pair of sentences, the algorithm starts by looking for similarity between the root nodes using lexical information and then proceeds looking for further couplings involving the immediate lexical nodes (children). In order to maximize structural isomorphism, the algorithm always backtracks to look for coupling of large lexical relationships. In this way, each mapping that is tried involve the estimation of its cost so that the best cost mapping can be selected as the optimal isomorphism.

How much content has to overlap between a particular pair of texts so we are able to recognize in such sentences an entailment relationship? Since we are not targeting lexical similarity, an overlapping word or two is not taken as sufficient criterion for entailment recognition. Rather, we are interested in structural isomorphism so our classification criterion relies on clause overlaps.

Table 2. Greedy search mechanism



Only in this way, the pair of texts are said to be inferred one (H) from another one (T), no matter how much they otherwise differ.

For example, the dependency trees in Table 2 allow us to observe two clauses, one within another one, which the algorithm must try to match. As we are looking for clause overlaps, from the whole set of semantic relations we are just interested in few of them. So the algorithm starts by focusing on semantic relations that allow us to spot the event and its corresponding arguments. In this particular case, the algorithm starts by focussing on the bigger clause denoted by *said*, and then on the clause denoted by *set*.

The dependencies trees in Table 2 also permit us to observe why a greedy search mechanism is necessary for entailment recognition. In this particular case, the algorithm backtracks when after focusing on the bigger clause denoted by *said*, it then proceeds with the clause denoted by *triggered*. We can appreciate how in this way the algorithm searches for large lexical relationships when

Table 3. Accuracy obtained for each dataset

Dataset	Logistic	SVM
QA	0.48	0.53
IR	0.60	0.61
PP	0.68	0.60
IE	0.58	0.58

starts looking for similarity among the arguments of *said* and then among the arguments of *triggered*.

Secondly, for our inference mechanism we observed two linear learning models: logistic regression and SVM. As we know, logistic regression is mainly used for binary classification so making use of this model to recognize textual entailment sounds appropriated. On the other hand, SVM is a hyperplane classifier which has proved to be a useful approach to cope with natural text tasks i.e. it has been shown by researchers that SVM is one of the most accurate algorithms for text classification [4].

Through the application of the learning models we then focus on the estimation of the vector weights required in the entailment decision. Since the purpose of our experimental methods is to maximize structural isomorphism, we estimated the similarity between the text and hypothesis text according to how much of the hypothesis' dependency tree is included in the text. In fact, the size of the feature vector represents the search depth of the included subtree carried out by the search mechanism. In this way, the estimation of the vector weights allows us to observe the relevancy of each particular feature f_i with which the recognition of textual entailment is estimated.

4 Experimental Evaluation

Experiments were carried out using Weka, a well known collection of machine learning algorithms for data mining tasks [5]. Also, the experimentation was conducted using an evaluation method known as stratified k-fold cross validation.

An important point to be considered for experiments was the size of the feature vector. From a pragmatic perspective, it was essential to determine how so deep to search for similarity. We tried different sizes for the vector and the best results were obtained with four features, that is, the first four levels of each dependencies tree were used for our classification process.

We selected four datasets for the experimentation where accuracy was applied as the main measure. The accuracy obtained for each dataset is shown in Table 3. As it can be appreciated, the SVM learning model produced the best results on two datasets: QA and IR, whereas IE showed no difference at all. Surprisingly, the logistic model produced far better results on the PP dataset.

The main point to highlight is to observe how some data sets can be linearly separated whereas others require nonlinear decision boundaries. In fact, QA dataset is the most notorious case which exhibited the necessity of dealing

Table 4. Average number of dependencies for each dataset

Dataset	Dependencies
QA	15
IR	12
PP	8
IE	13

with nonlinearly separable data. On the other hand, the results obtained on PP showed how this particular dataset was linearly separated by making use of a hyperplane classifier.

Also, an elemental linguistic analysis was developed for the reflection of this experimentation. Firstly, from the linguistic transformations such as nominalization, passivisation and negation, it was the negation transformation which played a crucial role on the PP dataset. For illustration purposes we make use of the following T-H instance represented by the following pair of sentences:

T Those accounts were not officially confirmed by the Uzbek or American governments.
H The Uzbek or American governments confirmed those accounts.

This particular T-H instance exhibits high structural isomorphism. Nevertheless, the negation transformation of T should have been considered, otherwise a poor performance would have been reported. On the other hand, the negation transformation makes no difference on the rest of the datasets analyzed.

Another observation was the difference in the number of grammatical relationships (dependencies) between each of the textual pairs. Table 4 shows the average number of dependencies corresponding to each dataset. From Tables 3 and 4 we can see a correlation between the number of dependencies and the learning model for decision boundaries. The use of nonlinearly learning models seems to be more appropriated for datasets with a higher difference in number of dependencies, whereas a linear hyperplane classifier seems to be a plausible option for pair of texts whose difference in the number of dependencies is not significant. In short, this observation highlights the necessity of analyzing the nature of the dataset in order to spot relevant features across applications [6].

5 Conclusions and Future Work

In this paper we define a measure for textual entailment recognition (RTE) based on structural isomorphism theory applied to lexical dependency information. We describe the experiments in terms of a feature set by making use of learning models such as linear (logistic regression) and nonlinear learning models (SVM) to estimate measure's parameters. The results of the experimentation conducted showed how a model constructed around lexical relationships is a plausible alternative for textual entailment recognition.

Future work is focused in trying other alternatives for improving performance in QA dataset, taking into account either different linguistic features such as those experimented by Rodrigo [7] in his study of the effect of named entities in the entailment recognition task, or the inference rules suggested by Lin and Pantel [8]. Also, we plant to try other alternatives for recognizing false entailment. For example, Snow et al. [9] suggests a set of heuristics for alignment such as negation mismatch and indirect relations between two aligned verb nodes.

Acknowledgment

I wish to thank Professor Yorick Wilks for his support during my summer research stay with the Sheffield NLP group to explore the topic of textual entailment. This work has been supported by the Consejo Nacional de Ciencia y Tecnología (CONACYT) within the Program for the Institutional Consolidation under the repatriation conditions of the application submitted at 2006.

References

1. Dagan, I., Glickman, O.: Probabilistic textual entailment: Generic applied modeling of language variability. In: Pascal workshop on Learning Methods for Texts Understanding and Mining, Grenoble, France, pp. 26–29 (2004)
2. Pazienza, M., Pennacchiotti, M., Zanzotto, F.: Textual entailment as syntactic graph distance: a rule based and a svm based approach. In: Pascal Challenges Workshop, Southampton, UK (2005)
3. Matsumoto, Y., Ishimoto, H., Utsuro, T.: Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In: 31st Annual Meeting of the Association for Computational Linguistic, pp. 23–30 (1993)
4. Thorsten, J.: Text categorization with support vector machines: Learning with many relevant features. In: European Conference on Machine Learning (1998)
5. Witten, I., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
6. Boss, J., Markert, K.: Recognising textual entailment with logical inference. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 628–635 (2005)
7. Rodrigo, A., Peñas, A., Herrera, J., Verdejo, F.: Experiments of uned at the third recognizing textual entailment challenge. In: Proceedings of the Workshop on Textual Entailment and Paraphrasing, Prague, pp. 89–94 (2007)
8. Lin, D., Pantel, P.: Discovery of inference rules for question-answering. Natural Language Engineering 7, 343–360 (2001)
9. Snow, R., Vanderwende, L., Menezes, A.: Effectively using syntax for recognizing false entailment. In: Proceedings of the HLT/NAACL, pp. 33–40 (2006)

On-Line Modeling Via Fuzzy Support Vector Machines

Julio César Tovar and Wen Yu

Departamento de Control Automático, CINVESTAV-IPN
A.P. 14-740, Av.IPN 2508, México D.F., 07360, México
yuw@ctrl.cinvestav.mx

Abstract. This paper describes a novel nonlinear modeling approach by on-line clustering, fuzzy rules and support vector machine. Structure identification is realized by an on-line clustering method and fuzzy support vector machines, the fuzzy rules are generated automatically. Time-varying learning rates are applied for updating the membership functions of the fuzzy rules. Finally, the upper bounds of the modeling errors are proven.

1 Introduction

Both neural networks and fuzzy logic are universal estimators, they can approximate any nonlinear function to any prescribed accuracy, provided that sufficient hidden neurons and fuzzy rules are available [2] [12]. The fusion of some intelligent technologies with fuzzy systems seems to be very effective for nonlinear systems modeling. Resent results show that neural networks and fuzzy logic seem to be very effective to identify a wide class of complex nonlinear systems when we have no complete model information, or even when we consider the controlled plant as a black box. Fuzzy modeling for nonlinear systems is based on a set of IF-THEN rules which use linguistic propositions of human thinking. The key problem of the fuzzy modeling is extraction of the fuzzy rules, it can be divided into two classes [11]: 1) obtaining fuzzy rules from experts, 2) obtaining fuzzy rules automatically from observed data. The expert method uses the un-bias criterion [16] and the trial-and-error technique, it can only be applied off-line.

The process of fuzzy rule extraction for nonlinear systems modeling is called structure identification. A common method is to partition the input and the output data, also called fuzzy grid [10]. Most of structure identification approaches are based on off-line data clustering, such as fuzzy C-means clustering, mountain clustering [13], and subtractive clustering [5]. These approaches require that the data is ready before the modeling. There are a few of on-line clustering methods in the literature. A combination of on-line clustering and genetic algorithm for fuzzy systems is proposed in [9]. In [17] the input space was automatically partitioned into fuzzy subsets by adaptive resonance theory. On-line clustering with a recursively calculated spatial proximity measure was given in [1]. There is one weakness for the above on-line clustering methods: the partitioning of the input

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 220–229, 2008.

(precondition) and the output (consequent) do not take into account time mark. They use whole data to train each rule. In this paper, a novel on-line clustering approach is proposed. The time relationship in the input and the output spaces is considered.

Besides clustering approaches, fuzzy rule extraction can also be realized by neural networks method [10], genetic algorithms [16], Singular-Value Decomposition (SVD-QR) [4] and support vector machines (SVM) technique [7]. SVM was first used for solving pattern classification problem. Vapnik defined it as a structure risk minimization which minimizes the upper bound of the modeling error. The basic idea of SVM modeling is to map the inputs into higher dimensional feature space, then solve quadratic programming (QP) with an appropriate cost function [14]. SVM has one important property: the solution vector is sparse. Only the non-zero solutions which are called support vectors are useful for the model. In this paper, we use the support vectors to extract the fuzzy rules in each cluster after the on-line clustering.

In this paper, a three phases fuzzy support vector machines with on-line clustering is proposed. First, we use an on-line clustering method which divides the input and the output data into several clusters in the same temporal interval, so the structure of fuzzy systems is automatically established. Second, fuzzy SVM is applied to generate support vectors in each cluster. With these support vectors, fuzzy rules are constructed. Here we use two fuzzy techniques to modify the standard SVM, the kernel is changed as fuzzy membership function and a fuzzy factor is added to the performance index of SVM, and the corresponding fuzzy systems are made. The fuzzy SVM provides an adaptive local representation for SVM, and this takes advantages of some properties of a fuzzy system, such as adaptive learning and economic structure [8].

2 On-Line Clustering for the Input/Output Data

The following discrete-time nonlinear system will be identified by our fuzzy modeling method

$$y(k) = f\left[\mathbf{x}\left(k\right), k\right] \tag{1}$$

where

$$\mathbf{x}\left(k\right) = \left[y\left(k-1\right), y\left(k-2\right), \cdots u\left(k-d\right), u\left(k-d-1\right), \cdots\right]^T \tag{2}$$

$f\left(\cdot\right)$ is an unknown slow time-varying nonlinear function, representing the plant dynamics, $u\left(k\right)$ and $y\left(k\right)$ are measurable scalar input and output of the nonlinear plant, d is time delay, $\mathbf{x}\left(k\right) \in \Re^n$ can be regarded as new input to the nonlinear function $f\left(\cdot\right)$, it is a NARMAX model [3].

The objective of structure identification is to partition the input and the output data $\left[y\left(k\right), \mathbf{x}\left(k\right)\right]$ of the nonlinear system (1) and extract fuzzy rules. We use the following simple example to explain the importance of the on-line clustering proposed in this paper. We consider a nonlinear function as (1).

The nonlinear mapping from $x\left(k\right)$ to $y\left(k\right)$ By the normal on-line clustering method proposed in [1] the input and output may be partitioned into 4 clusters,

$A_1 - A_4$ and $B_1 - B_4$. From there, 4 rules can be formulated as "IF $x(k)$ is A_j THEN $y(k)$ is B_j", $j = 1 \cdots 4$. For example, the 3rd rule is "IF $x(k)$ is A_3 THEN $y(k)$ is B_3", this fuzzy rule does not represent relation (**??**) well, because the precondition $x(k)$ and the consequent $y(k)$ do not occur at the same time for this rule.

In this paper, the input and the output spaces are partitioned in the same temporal interval. There are two reasons. Firstly, nonlinear system modeling is to find a suitable mapping between the input and the output. Only when the input and the output of the fuzzy system occur in the same time interval, the fuzzy rules can correspond to the nonlinear mapping. Secondly, we consider an on-line modeling approach. When a new cluster (or a new rule) is created, we do not want to use all data to train it as in [1]. If the data have time marks, only the data in the corresponding time interval should be used. So clustering with temporal intervals will simplify parameter identification and help to make the task of on-line modeling.

The basic idea of the proposed on-line clustering scheme is, if the distance from a point to the center is less than a required length, the point is in this cluster. When a new data comes, the center and the cluster should be changed according to the new data. We give the following algorithm. The Euclidean distance at time k is defined as

$$d_{k,\mathbf{x}} = \left(\sum_{i=1}^{n} \left[\frac{x_i(k) - \overline{x}_i^j}{x_{i,\max} - x_{i,\min}} \right]^2 \right)^{1/2}, \quad d_{k,y} = \left| \frac{y(k) - \overline{y}^j}{y_{\max} - y_{\min}} \right| \tag{3}$$

$$d_k = \alpha d_{k,\mathbf{x}} + (1 - \alpha) d_{k,y} \tag{4}$$

where $x_{i,\max} = \max_k \{x_i(k)\}$, $x_{i,\min} = \min_k \{x_i(k)\}$, $y_{\max} = \max_k \{y(k)\}$, $y_{\min} = \min_k \{y(k)\}$, $\overline{x}^j(k)$ and $\overline{y}^j(k)$ are the centers of jth cluster, i is the dimension of input, α is a positive factor, normally we can choose $\alpha = \frac{1}{2}$. For the cluster j, the centers are updated as

$$\overline{x}_i^j = \frac{1}{l_2^j - l_1^j + 1} \sum_{l=l_1^j}^{l_2^j} x_i(l), \quad \overline{y}^j = \frac{1}{l_2^j - l_1^j + 1} \sum_{l=l_1^j}^{l_2^j} y(l) \tag{5}$$

where l_1^j is the first time index of the cluster j, l_2^j is the last time index of the cluster j. The length of cluster j is $m^j = l_2^j - l_1^j + 1$. The time interval of cluster j is $\left[l_1^j, l_2^j \right]$. The process of the structure identification can be formed as the following steps,

1. For the first data G_1, $k = 1$. $y(1), x_i(1)$ are the centers of the first cluster, $\overline{x}_i^1 = x_i(1)$, $\overline{y}^j = y(1)$, $l_1^j = l_2^j = 1$, $j = 1$.
2. If a new data $[y(k), x_i(k)]$ comes, increment l_2^j by 1, we use (5) and (3) to calculate d_k. If no any new data comes, goto 5.
3. If $d_k \leq L$ then $[y(k), x_i(k)]$ is still in cluster G_j, goto 2

4. If $d_k > L$ then $[y(k), x_i(k)]$ is in a new cluster, increment j by 1, the center of the G_j is $\overline{x}_i^j = x_i(k)$, $\overline{y}^j = y(k)$, $l_1^j = l_2^j = k$, goto 2
5. Check the distances between all centers $\overline{\mathbf{x}}^j$, $\overline{y}^j$, if $\sum_{i=1}^{n} [\overline{x}_i^p - \overline{x}_i^q]^2 + |\overline{y}^p - \overline{y}^q| \leq L$, the two cluster G_p and G_q are combine into one cluster.

There are two design parameters α and L. α is regarded as the weight on the input space. If the input dominates the dynamic property, we should increase α. Usually we select $\alpha = 0.5$ such that the input and the output have equal influence. If we let $\alpha = 1$, it becomes the normal on-line clustering [1][9][17]. L is the threshold of creating new rules, it is the lowest possible value of similarity. How to choose the user-defined threshold is a trade-off problem. If the threshold value L is too small, there are many clusters and some of them will be singletons. Conversely, if the threshold L is too large, many objects that are not similar may be partitioned in the same cluster. Since $d_k = \alpha d_{k,\mathbf{x}} + (1-\alpha) d_{k,y}$, so $d_{\max} = \alpha \|\mathbf{x}_{\max} - \mathbf{x}_{\min}\| + (1-\alpha) \|y_{\max} - y_{\min}\|$. If we want several clusters, we should let $L < d_{\max}$, otherwise there is only one cluster.

3 Fuzzy Rules Extraction by Fuzzy Support Vector Machines

A SVM can separate the data into two classes with a maximum margin hyperplane [7]. If the training is detachable for the hyperplane, the function is chosen as $h(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b$. The margin is defined as the minimum distance from a sample to the surface of resolution. We can measure this margin by the longitude of the vector $\mathbf{w}$. In this way the near points to the hyperplane satisfies $|(\mathbf{w} \cdot \mathbf{x}) + b| = 1$.

In this paper we use SVM for function estimation. In order to find support vectors in the cluster j, we use the input/output data $[\mathbf{x}_k, y(k)]$, $k \in \left[l_1^j, l_2^j\right]$ to approximate a nonlinear function. Consider a nonlinear function regression $y = f(\mathbf{x})$ can be estimated as the following form,

$$h(\mathbf{x}_k) = \mathbf{w}^T \varphi(\mathbf{x}_k) + b \tag{6}$$

where $\mathbf{w}$ is the weight vector, $\varphi(\mathbf{x}_k)$ is a known nonlinear function, b is a threshold, $\mathbf{x}_k$ is the input vector at time k, Kernel trick is defined as$K(\mathbf{x}, \mathbf{x}_k) = \varphi(\mathbf{x})^T \varphi(\mathbf{x}_k)$, $\mathbf{x}$ is the input vector at any time

3.1 Fuzzy Kernel

There are many possible choices for the kernel $K(\mathbf{x}, \mathbf{x}_k)$, we only require $K(\mathbf{x}, \mathbf{x}_k)$ satisfies the Mercer condition [7]. For example linear kernel $K(\mathbf{x}, \mathbf{x}_k) = \mathbf{x}_k^T \mathbf{x}$, MLP kernel $K(\mathbf{x}, \mathbf{x}_k) = \tanh\left(a_1 \mathbf{x}_k^T \mathbf{x} + a_2\right)$, a_1 and a_2 are constants. RBF kernel

$K(\mathbf{x}, \mathbf{x}_k) = \exp\left(-\|\mathbf{x} - \mathbf{x}_k\|^2 / \sigma^2\right)$. In this paper, we use fuzzy kernel which is defined as

$$K(\mathbf{x}, \mathbf{x}_k) = \begin{cases} \prod_{i=1}^{M} u_i(\mathbf{x}_k) \cdot u_i(\mathbf{x}_i) & \mathbf{x}_k \text{ and } \mathbf{x}_i \text{ are both in the } jth \text{ cluster} \\ 0 & \text{otherwise} \end{cases}$$

where M is total time, $u_i(\mathbf{x}_k)$ is the membership function.

Let the training set be $S = \{(\mathbf{x}_1, y(1)), (\mathbf{x}_2, y(2)), \ldots, (\mathbf{x}_M, y(M))\}$. Assume the training samples are partitioned into l clusters. We can group the training samples into l clusters as follows

$$\begin{array}{rl} \text{cluster } 1 & = \left\{\left(\mathbf{x}_1^1, y_1^1\right), \ldots, \left(\mathbf{x}_{k_1}^1, y_{k_1}^1\right)\right\} \\ \text{cluster } 2 & = \left\{\left(\mathbf{x}_1^2, y_1^2\right), \ldots, \left(\mathbf{x}_{k_2}^2, y_{k_2}^2\right)\right\} \\ & \vdots \\ \text{cluster } l & = \left\{\left(\mathbf{x}_1^l, y_1^l\right), \ldots, \left(\mathbf{x}_{k_l}^l, y_{k_l}^l\right)\right\} \end{array} \tag{7}$$

where $\mathbf{x}_{k_1}^g$ and $y_{k_1}^g$ are the first input/output of the cluster g, $\mathbf{x}_{k_g}^g$ $y_{k_g}^g$ are the last input/output of the cluster g, $g = 1, 2, \ldots, l$ is the number of points belonging to the gth cluster, so that we have $\sum_{g=1}^{l} k_g = M$.

Then the fuzzy kernel can be calculated by using the training set in (7), and the resulting kernel matrix K can be rewritten as the following form [6]:

$$K = \begin{bmatrix} \mathbf{K}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{K}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{K}_l \end{bmatrix} \in R^{M \times M} \tag{8}$$

How to choose the membership function $u_i(\mathbf{x}_k)$ is another problem. Gaussian function and triangle function are most popular functions for the membership function of fuzzy systems. When $u_i(\mathbf{x}_k)$ is Gaussian function, the kernel function is

$$G = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_i) = \exp\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{9}$$

and the fuzzy kernel is

$$\mathbf{K}_g = \begin{bmatrix} G\left(\mathbf{x}_{k_1}^g, \mathbf{x}_{k_1}^g\right) & G\left(\mathbf{x}_{k_1}^g, \mathbf{x}_{k_1+1}^g\right) & \cdots G\left(\mathbf{x}_{k_1}^g, \mathbf{x}_{k_g}^g\right) \\ G\left(\mathbf{x}_{k_1+1}^g, \mathbf{x}_{k_1}^g\right) & & \vdots \\ \vdots & \ddots & \cdots G\left(\mathbf{x}_{k_g-1}^g, \mathbf{x}_{k_g}^g\right) \\ G\left(\mathbf{x}_{k_g}^g, \mathbf{x}_{k_1}^g\right) & \cdots G\left(\mathbf{x}_{k_g}^g, \mathbf{x}_{k_g-1}^g\right) & \cdots G\left(\mathbf{x}_{k_g}^g, \mathbf{x}_{k_g}^g\right) \end{bmatrix} \in R^{k_g \times k_g} \tag{10}$$

3.2 Fuzzy Support Vector Machines

The normal cost function of nonlinear regression is defined as

$$R_{emp}(\theta) = \frac{1}{N}\sum_{k=l_1^j}^{l_2^j} \left| y_k - \left(\mathbf{w}^T \varphi(\mathbf{x}_k) + b\right)\right|_\epsilon$$

where ϵ is the Vapnik's insensitive loss function, defined as

$$|y_k - f(\mathbf{x})|_\epsilon = \begin{cases} 0 & |y_k - f(\mathbf{x})| \le \epsilon \\ |y_k - f(\mathbf{x})| - \epsilon & \text{otherwise} \end{cases}$$

ϵ can be regarded as the accuracy of approximation.

In this paper, we introduce a fuzzy factor s_i to the above performance index [6]. Here $\sigma \le s_i \le 1$, where σ is a sufficient small positive number, s_i denotes the important degree of sample x_i for learning the optimal hyperplane in SVM. We select s_i as bell-shape function (11).

$$s_i = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \tag{11}$$

The optimal hyperplane problem is regarded as the solution to the following problem (primal problem)

$$\begin{aligned} &\min J_p = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{k=1}^{n} s_k\xi_k \\ &\text{subject to } \left| y_k - \left(\mathbf{w}^T\varphi(\mathbf{x}_k) + b\right)\right| \le \epsilon \end{aligned} \tag{12}$$

where $\xi_k \ge 0$, $k = 1, 2, \ldots, n$, C is constant, the smaller s_k is, the smaller the effect that the sample $\mathbf{x}_k$ to optimal hyperplane. If ϵ is too small, certain points will be outside of this ϵ tube. Therefore additional slack variables ξ_i are introduced (it is somewhat similar to the classifier case with overlapping). To solve the optimal problem above, we can construct the following Lagrangian function with inequality constraint and slack variables

$$\begin{aligned} &L(w, b, \xi, a, \beta) = \tfrac{1}{2}\|w\|^2 + C\sum_{k=1}^{n} s_k\xi_k \\ &-\sum_{i=1}^{n}\alpha_k\left(y_k - (w\cdot\varphi(\mathbf{x}_k) + b) - 1 + \xi_k\right) - \sum_{k=1}^{n}\beta_k\xi_k \end{aligned} \tag{13}$$

where α_k, $\beta_k \ge 0$ are the Lagrangian multipliers. The primal problem (12) can be transformed into the dual problem by differentiating L with respect to w, ξ and b.

From Kuhn-Tucker conditions, necessary and sufficient conditions for the optimal (w^*, α^*, β^*) are

$$\begin{aligned}
&\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial w} = w - \sum_{k=1}^{n} \alpha_k y_k \varphi(\mathbf{x}_k) = 0 \\
&\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial b} = -\sum_{i=1}^{n} \alpha_k y_k = 0 \\
&\frac{\partial Q(w,b,\xi,\alpha,\beta)}{\partial \xi_k} = s_k C - \alpha_k - \beta_k = 0 \\
&\left| y_k - \left(\mathbf{w}^T \varphi(\mathbf{x}_k) + b \right) \right| \leq \epsilon, \ \alpha_k, \beta_k \geq 0
\end{aligned} \tag{14}$$

and

$$\alpha_k^* \left\{ y_k \left[\mathbf{w}^T \varphi(\mathbf{x}_k) + b \right] - 1 + \xi_k \right\} = 0 \tag{15}$$

Apply these conditions into the Lagrangian (13), the primal problem (12) can be changed into the following dual problem (quadratic programming)

$$\begin{aligned}
&\max W(\alpha) = \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\
&\text{subject to} \sum_{i=1}^{n} \alpha_i y_i = 0, \qquad \alpha_i \geq 0, K(x_i, x_j) \geq 0
\end{aligned} \tag{16}$$

where $0 \leq \alpha_i \leq s_i C$, $i = 1, 2, \ldots, n$, $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ is fuzzy kernel.

By standard QP software package we obtain the solution α_k^*. b can be solved by (15). The resulting function is

$$f(\mathbf{x}) = \sum_{k=l_1^j}^{l_2^j} (\alpha_k - \alpha_k^*) K(\mathbf{x}_k, \mathbf{x}) + b$$

Since many $\alpha_k^* = 0$, the solution vector is sparse, the sum should be taken only over the non-zero α_k (support vector), so the final result is

$$f(\mathbf{x}) = \sum_{k=1}^{sv_j} (\alpha_k - \alpha_k^*) K(\mathbf{x}_k, \mathbf{x}) + b \tag{17}$$

where sv_j is the number of the support vectors, $sv_j \ll \left(l_2^j - l_1^j + 1 \right)$. We define the positions of the support vectors as $[x_i^*, y_i^*]$, $i = 1 \cdots sv_j$.

The point $\mathbf{x}_k$ with the corresponding $0 \leq \alpha_k \leq s_k C$ is called a support vector. For fuzzy SVM, there are two types of support vectors, the one corresponding to $0 \leq \alpha_k \leq s_k C$ lies on the margin of the hyperplane, and the one corresponding to $\alpha_i = s_i C$ is mis-classified. So the points with the same value of α_k in fuzzy SVM may indicate a different type of support vector in fuzzy SVM due to the factors s_i.

From (17) we know the support vectors are enough to represent the nonlinearity in each cluster. We use these support vectors to construct fuzzy rules. For

the cluster j, we extract fuzzy product rules in the following form (Mamdani fuzzy model)

$$\mathrm{R}^j\text{: IF } x_1(k) \text{ is } A_1^j \text{ and } x_2(k) \text{ is } A_2^j \text{ and } \cdots x_n(k) \text{ is } A_n^j \text{ THEN } y(k) \text{ is } B^j$$

here $j = 1 \cdots sv_j$, $A_1^j, \cdots A_n^j$ and B^j are standard fuzzy sets. We use sv_j fuzzy IF-THEN rules to perform a mapping from an input vector $\mathbf{x} = [x_1 \cdots x_n] \in \Re^n$ to an output $y(k)$. The Gaussian membership function is

$$\mu_i(x_j) = \exp\left(-\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2}\right) \tag{18}$$

where i is the condition in the part "IF", $i = 1 \cdots n$, j is rule number, $j = 1 \cdots sv_j$. If $[x_i^*, y_i^*]$ is the point of the support vector, we let $\mathbf{x}^*(k)$ and $y^*(k)$ as the center of the Gaussian functions.

From [18] we know, by using product inference, center-average and singleton fuzzifier, the output of the fuzzy system in cluster j can be expressed as

$$\hat{y} = \left(\sum_{j=1}^{sv_j} w_j \left[\prod_{i=1}^{n} \mu_{A_i^j}\right]\right) / \left(\sum_{j=1}^{sv_j} \left[\prod_{i=1}^{n} \mu_{A_i^j}\right]\right) \tag{19}$$

where $\mu_{A_i^j}$ is the membership functions of the fuzzy sets A_i^j, w_j is the point at which $\mu_{B_j} = 1$. If we define

$$\phi_i = \prod_{j=1}^{n} \mu_{A_{ji}} / \sum_{i=1}^{sv_j} \prod_{j=1}^{n} \mu_{A_{ji}}$$

(19) can be expressed in matrix form

$$\hat{y}(k) = \mathbf{W}(k)\, \Phi[\mathbf{x}(k)] \tag{20}$$

where the parameter $\mathbf{W}(k) = \left[W_1 \cdots W_{sv_j}\right] \in R^{sv_j}$, data vector $\Phi[\mathbf{x}(k)] = \left[\phi_1 \cdots \phi_{sv_j}\right]^T \in R^{sv_j \times 1}$.

Now, each cluster has a fuzzy model. This model is valid in the time intervals of the cluster. It is a piece-wise nonlinear function. For example, in cluster 3 the time interval is $k \in \left[l_1^3, l_2^3\right]$, the fuzzy model for the cluster is $\hat{y} = f_3(\cdot)$. This means that

$$\hat{y} = f_3(\mathbf{x}(k)), \quad k \in \left[l_1^3, l_2^3\right]$$

We can use the idea of Takagi-Sugeno model to combine these local models into a global model. We define the following fuzzy rules which are corresponded to each cluster

$$\mathrm{R}^j\text{: IF } x_{l_1^j} \le x_1 \le x_{l_2^j}, \text{ and } \cdots x_{l_1^j} \le x_n \le x_{l_2^j} \text{ THEN } \hat{y}(k) = f_j[\mathbf{x}(k)]$$

where $j = 1 \cdots p$, p is the cluster number by the on-line clustering. The membership functions for x_i is defined as a trapezoidal function

$$\mu_{A_i^j}(x) = \begin{cases} \frac{x-a}{x_{l_1^j}-a} & x \in \left[a, x_{l_1^j}\right) \\ 1 & x \in \left[x_{l_1^j}, x_{l_2^j}\right] \\ \frac{x-b}{x_{l_2^j}-b} & x \in \left(x_{l_2^j}, b\right] \\ 0 & otherwise \end{cases} \quad \begin{array}{l} a = x_{l_1^j} - \frac{1}{4}x_{l_1^j} \\ b = x_{l_2^j} + \frac{1}{4}x_{l_2^j} \end{array}$$

The final fuzzy model is

$$\hat{y} = \left(\sum_{j=1}^{p} f_j\left[\mathbf{x}(k)\right]\left[\prod_{i=1}^{n}\mu_{A_i^j}\right]\right) / \left(\sum_{j=1}^{p}\left[\prod_{i=1}^{n}\mu_{A_i^j}\right]\right) \tag{21}$$

4 Conclusions

In this paper we propose an efficient approach for nonlinear system modeling using fuzzy rules. Several techniques are combined for the new approach. First we propose an on-line clustering method which divides the input/output data into several clusters in a same temporal interval. Then we apply fuzzy support vector machines which generate support vectors in each cluster. With these support vectors, fuzzy rules are constructed and the corresponded fuzzy system is made.

References

1. Angelov, P.: An approach for fuzzy rule-base adaptation using on-line clustering. International Journal of Approximate Reasoning 35(3), 275–289 (2004)
2. Brown, M., Harris, C.J.: Neurofuzzy Aadaptive Modelling and Control. Prentice Hall, New York (1994)
3. Chen, S., Billings, S.A.: Neural networks for nonlinear system modelling and identification. Int. J. Control 56(2), 319–346 (1992)
4. Chiang, J.-H., Hao, P.-Y.: Suuport Vector Learning Mechanism for Fuzzy Rule-Based Modeling: A New Approach. IEEE Transactions on Fuzzy Systems 12(1) (2004)
5. Chiu, S.L.: Fuzzy Model Identification based on cluster estimation. Journal of Intelligent and Fuzzy Systems 2(3) (1994)
6. Lin, C.-F., Wang, S.-D.: Fuzzy Support Vector Machines. IEEE Transactions on Neural Networks 13(2), 464–471 (2002)
7. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge Univ. Press, Cambridge (2000)
8. Hong-Sen, Xu, D.: An Approach to Estimating Product Desing Time Based on Fuzzy v-Support Vector Machine. IEEE Transactions on Neural Networks 18(3), 721–731 (2007)
9. Juang, C.F.: Combination of on-line clustering and Q-value based GA for reinforcement fuzzy system design. IEEE Transactions on Fuzzy Systems 13(3), 289–302 (2005)

10. Jang, J.S.: ANFIS: Adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23, 665–685 (1993)
11. Leski, J.M.: TSK-Fuzzy Modeling Based on ε-Insensitive Learning. IEEE Trans. on Fuzzy System 13(2), 181–193 (2005)
12. Lin, C.-T., Lee, C.-.S.G.: Neuralnetwork-based fuzzy logic control and decision system. IEEE Trans. Comput. 40, 1320–1336 (1991)
13. Mitra, S., Hayashi, Y.: Neuro–fuzzy rule generation: survey in soft computing framework. IEEE Transactions on Neural Networks 11(3), 748–769 (2000)
14. Mueller, K.-R., Mika, S., Rasch, G., Tsuda, K., Scholkopf, B.: An Introduction to Kernel-Based Learning Algorithm. IEEE Trans. Neural Networks 12(2) (2001)
15. Narendra, K.S., Mukhopadhyay, S.: Adaptive Control Using Neural Networks and Approximate Models. IEEE Trans. Neural Networks 8(3), 475–485 (1997)
16. Rivals, I., Personnaz, L.: Neural-network construction and selection in nonlinear modeling. IEEE Transactions on Neural Networks 14(4), 804–820 (2003)
17. Tzafestas, S.G., Zikidis, K.C.: NeuroFAST: On-line neuro-fuzzy ART-based structure and parameter learning TSK model. IEEE Transactions on Systems, Man and Cybernetics, Part B 31(5), 797–803 (2001)
18. Wang, L.X.: Adaptive Fuzzy Systems and Control. Prentice-Hall, Englewood Cliffs (1994)
19. Wang, C.H., Liu, H.L., Lin, C.T.: Dynamic optimal learning rates of a certain class of fuzzy neural networks and its applications with genetic algorithm. IEEE Trans. Syst., Man, Cybern. B 31, 467–475 (2001)

Reinforcement Learning with Markov Logic Networks

Weiwei Wang, Yang Gao, Xingguo Chen, and Shen Ge

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, PRC
elegate@gmail.com

Abstract. In this paper, we propose a method to combine reinforcement learning (RL) and Markov logic networks (MLN). RL usually does not consider the inherent relations or logical connections of the features. Markov logic networks combines first-order logic and graphical model and it can represent a wide variety of knowledge compactly and abstractly. We propose a new method, reinforcement learning algorithm with Markov logic networks (RLMLN), to deal with many difficult problems in RL which have much prior knowledge to employ and need some relational representation of states. With RLMLN, prior knowledge can be easily introduced to the learning systems and the learning process will become more efficient. Experiments on blocks world illustrate that RLMLN is a promising method.

1 Introduction

State representation is a critical task in RL. Function approximation is an approach to dealing with high-dimension tasks. However, RL usually does not consider inherent relations or connections of the features. Otherwise we need to introduce additional features to represent such connections. Therefore, we need a high level relational and abstract representation in RL for real world problems where there are enormous state spaces.

Relational reinforcement learning (RRL) is concerned with upgrading the representation of RL methods to the first-order case, that is, reasoning and learning about objects and relations between objects ([1]). RRL is modeled by relational MDPs (RMDPs). For a detailed definition, please see ([1]). Recently, researchers have presented a lot of methods to solve RRL problems, which can be classified into three classes: model-free, partially modeling and model-based. Among these methods, many integrate first-order logic with traditional method and gain the ability to compactly and declaratively represent complex problems. For example, LOMDP ([2]) use clauses or formulas to partition state space and learn state values for those formulas.

Markov logic networks (MLN), proposed by Matthew Richardson and Pedro Domingos ([3]), attaches weights to first-order formulas and takes them as features of the network. It can be viewed as a template for generating ground Markov networks. In this way, MLN achieves the goal of combining graphical

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 230–242, 2008.

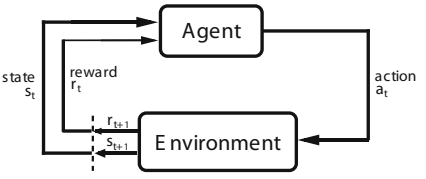


Fig. 1. The agent-environment interaction in reinforcement learning

model and first-order logic, so that it handles the complexity and uncertainty of the real world in a single framework. Recently, MLN has been used to deal with various kinds of problems. For example, Parag Singla and Pedro Domingos used it to do entity resolution ([4]), collective classification and so on. Transfer learning based on MLN ([5]) is also undertaken.

The above work inspires us to propose a reinforcement learning algorithm with Markov logic networks (RLMLN) to combine RL and MLN. In RLMLN, MLN does inference for the action queries and selects a best action, while RL uses the successive state, current state and the reward to update the weights of formulas in MLN. With RLMLN, we can compactly represent a state in RL for those problems which have enormous state space and need abstract state representation. Furthermore, we can easily introduce prior knowledge to a learning system. We apply RLMLN to the problem of blocks world ([6]) and the experimental results show that RLMLN is a promising method.

The rest of this paper is organized as follows. In sections 2 and 3, we briefly review reinforcement learning and Markov logic networks. Section 4 presents RLMLN. Section 5 gives the experiments on blocks world. Finally, section 6 concludes.

2 Reinforcement Learning

Reinforcement learning is learning what to do–how to map situations to actions–so as to maximize a numerical reward signal([7]). A reinforcement learning agent learns knowledge from interaction with the environment, as shown in Fig. 1. We normally describe a reinforcement learning problem using Markov decision processes(MDPs). A MDP is a four tuple $< S, A, T, R >$ consisting of a finite state space S, a finite action space A, a transition function $T : S \times A \times S \rightarrow \Re$, and a reward function $R : S \times A \rightarrow \Re$([7]). From a given state $s \in S$, a given action $a \in A(s)$ produces an expected reward of $R(s, a)$ and transitions to another state $s' \in S$ with probability $T(s, a, s')$. Monte-Carlo, TD(λ), Q-learning are some of the classical methods in RL.

3 Markov Logic Networks

Complexity and uncertainty is the nature of many real world applications ([8]). Logic mainly handles the former while statistical learning focuses on the latter.

However, a first-order KB has a hard constraint on the possible worlds because if one formula is violated, then the world will has zero probability. Markov logic soften the constraints by combining logic and probability with attached weights for first-order formulas: if a world violates a formula then it becomes less probable, but not impossible([9]).

3.1 Markov Logic

A Markov network is a model for the joint distribution of a set of variables $X = (X_1, X_2, \ldots, X_n) \in \chi$([10]). It consists of an undirected graph G and a set of potential functions ϕ_k. There's a node for each variable in the graph and the model has a potential function for each clique in the graph. The joint distribution represented by a Markov network is given below:

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}}) \ . \tag{1}$$

where $x_{\{k\}}$ is the state of the kth clique, Z=$\sum_{x \in \chi} \prod_k \phi_k(x_{\{k\}})$. For convenience, we often represent Markov networks as *log-linear* models:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_j w_j f_j(x) \right) \ . \tag{2}$$

where Z is a normalization factor, and $f_j(x)$ is one of the feature of state x. Markov logic introduces first-order logic to Markov networks. Recall that a first-order knowledge base is a set of formulas represented in first-order logic. Constants(e. g. , Jack, Johnson), variables, functions and predicates(e. g. , On(a, b), Has(x, y)) are the four types of symbols to construct these formulas. A grounded predicate is a predicate with its variables replaced by constants(e. g. , move(A, Floor), Has(John, computer)). A ground formula is a formula with no variables but constants. In Markov logic, a formula is a first-order logic formula attached with a weight. The weight of each formula reflects how strong a constraint the first-order formula has: the higher the weight, the greater the difference in probability between a world that satisfies the formula and one that does not. And a set of these formulas is called a Markov logic network or a MLN([8]).

The definition is given below([3]):

Definition 1. *A Markov logic network L consists of a set of pairs* (F_i, w_i)*, where* F_i *is a first-order logic formula and* w_i *is* F_i*'s weight(a real number).* $C = c_1, c_2, \ldots, c_{|C|}$ *is a set of constants. Markov network* $M_{L,C}$ *is defined as follows:*

1. $M_{L,C}$ *consists of binary nodes for each possible grounding of each predicate in L. The value of each node is* 1 *if the ground predicate is true,* 0 *otherwise.*
2. $M_{L,C}$ *contains one feature for each grounding of each formula* F_i *in L. The value of this feature is* 1 *if the ground formula is true,* 0 *otherwise. The weight of the feature* F_i *is* w_i*.*

Table 1. Example of MLN for blocks world

First-order logic	Clausal form	Weight
$\forall a, b, c, d$ on$(a, b) \wedge$ clear$(a) \wedge$ on(c, d)	$\neg$on$(a, b) \vee \neg$clear$(a) \vee \neg$on(c, d)	1.0
$\wedge$clear$(c) \Rightarrow$ move$(a, Floor)$	$\vee \neg$clear$(c) \vee$ move$(a, Floor)$	
$\forall a, b, c, d$ on$(a, b) \wedge$ clear$(a) \wedge$ on(c, d)	$\neg$on$(a, b) \vee \neg$clear$(a) \vee \neg$on(c, d)	1.0
$\wedge$cl$(c) \Rightarrow$ move$(x, Floor)$	$\vee \neg$clear$(c) \vee$ move(a, c)	

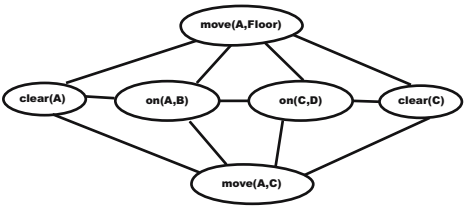


Fig. 2. Ground Markov network obtained by applying the formulas in Table 1 to constants `A`, `B`, `C` and `D`. And we delete nodes such as `move(A,A)`, `on(A,A)`.

A MLN can be used as a template to construct Markov networks and the resultant networks are called ground Markov networks. And the networks will vary when given different sets of constants. Table. 1 gives part of the formulas in a MLN for blocks world. An example of ground Markov networks for Table. 1 is shown in Fig. 2. The probability of world x specified by $M_{L,C}$ is given by:

$$P(X = x) = \frac{1}{Z}\exp\left(\sum_i w_i n_i(x)\right) = \frac{1}{Z}\prod_i \phi_i(x_{\{i\}})^{n_i(x)} \quad . \tag{3}$$

where $n_i(x)$ is the number of true groundings of F_i in x, $x_{\{i\}}$ is the state of the atoms appearing in F_i, and $\phi_i(x_{\{i\}}) = e^{w_i}$([3]).

3.2 Inference

Here we discuss the MCMC algorithm used in MLN for inference as our method is based on this inference technique. First we introduce the formula to calculate probability that formula F_1 holds given that F_2 does([3]).

$$\begin{aligned} P(F_1|F_2, L, C) &= P(F_1|F_2, M_{L,C}) \\ &= \frac{P(F_1 \wedge F_2|M_{L,C})}{P(F_2|M_{L,C})} \\ &= \frac{\sum_{x\in\chi_{F_1}\cap\chi_{F_2}} P(X = x|M_{L,C})}{\sum_{x\in\chi_{F_2}} P(X = x|M_{L,C})} \quad . \end{aligned} \tag{4}$$

where χ_{F_i} is the set of worlds where F_i holds, and $P(X|M_{L,C})$ is given by Eq.3.

$P(F_1|F_2, L, C)$ can be approximated using an MCMC algorithm that rejects all moves to states where F_2 does not hold, and counts the number of samples in which F_1 holds. In MCMC algorithm we have two steps to do for inference. First, we construct the minimal subset M of the ground Markov network required to compute $P(F_1|F_2, L, C)$. Second, we use Gibbs sampling to perform inference on this network. Gibbs sampling need a Markov blanket to sample one ground atom. The Markov blanket of a ground predicate is the set of ground predicates that appear in some grounding of a formula with it([3]). Given Markov blanket $B_l = b_l$, the probability of a ground predicate X_l is

$$P(X_l = x_l|B_l = b_l) = \frac{\exp(\sum_{f_i \in F_l} w_i f_i(X_l = x_l, B_l = b_l)))}{\exp(\sum_{f_i \in F_l} w_i f_i(X_l = 0, B_l = b_l)) + \exp(\sum_{f_i \in F_l} w_i f_i(X_l = 1, B_l = b_l))} \quad (5)$$

where F_l is the set of ground formulas that X_l appears in, and $f_i(X_l = x_l, B_l = b_l)$ is the value(0 or 1) of the feature corresponding to the ith ground formula when $X_l = x_l$ and $B_l = b_l$ ([3]).

For more detailed information about MLN, please refer to ([3]).

4 Reinforcement Learning Algorithm with Markov Logic Networks

Based on MLN, we propose our reinforcement learning algorithm with Markov logic networks. First we give a simple view of our method in Fig. 3. From the figure, we can see that given current state s, MLN helps calculate the probability of each action and randomly choose among the best ones, $a = \max_a P(a|s)$, and then next state s', reward r are received by RL agent from the environment. Using gradient-descent method, RL agent updates the weights vector $\vec{w}$ of MLN and then use the new weights to do inference. The inference algorithm we choose for MLN is MCMC as we mentioned before.

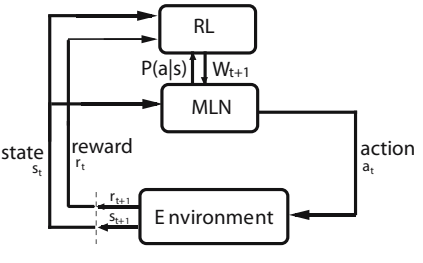


Fig. 3. A Framework of RLMLN

4.1 Update Weights with Gradient-Descent Methods

Gradient-descent method is a popular method used in RL and other fields of AI. The simple form is given below([7]):

$$\begin{aligned}\vec{\theta}_{t+1} &= \vec{\theta}_t + \frac{1}{2}\alpha \bigtriangledown_{\vec{\theta}_t} [V^\pi(s_t) - V_t(s_t)]^2 \\ &= \vec{\theta}_t + \alpha[V^\pi(s_t) - V_t(s_t)] \bigtriangledown_{\vec{\theta}_t} V_t(s_t) \ . \end{aligned} \tag{6}$$

where α is a positive step-size parameter, $V_t(s)$ is a smooth differentiable function of $\vec{\theta}_t$ for all $s \in \mathcal{S}$, $V^\pi(s_t)$ is the exact, correct value for each s_t, and $\bigtriangledown_{\vec{\theta}_t} V_t(s_t)$, for any function f, denotes the vector of partial derivatives. Normally, we can't get the exact value $V^\pi(s_t)$, so we use some approximation of it, denoted by v_t. Now we try to take MLN output as a representation of Q value and get a new formula:

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \alpha \left[r + \gamma \max_{a'} P(a'|s') - P(a|s) \right] \bigtriangledown_{\vec{\theta}_t} P(a|s) \ . \tag{7}$$

where $P(A = a|S = s) = \frac{1}{Z}\exp(\sum_j w_j f_j(x))$(Equ. 2). For simplicity of calculation of $\bigtriangledown_{\vec{\theta}_t} P(a|s)$, we adopt the ln form of $P(a|s)$. So we get

$$\frac{\partial}{\partial w_i} \ln P_w(A = a|S = s) = n_i(x) - \sum_{x'} P_w(X = x') n_i(x') \ . \tag{8}$$

where $n_i(x)$ is the number of true groundings in the data x for the ith formula. However, Equ. 8 is difficult for calculation. As mentioned before, we use MCMC algorithm in our method, so $P(a|s)$ is calculated using Equ. 5. In this way, we can get $\bigtriangledown_{\vec{\theta}_t} P(a|s)$ as below in a simple form:

$$\begin{aligned}&\frac{\partial}{\partial w_i} \ln P(X_l = x_l|B_l = b_l) \\ &= \sum_{f_i \in F_l} \left\{ f_i(X_l = x_l, B_l = b_l) - \sum_{x' \in 0,1} f_i(X_l = x', B_l = b_l) P(X_l = x'|B_l = b_l) \right\} \end{aligned} \tag{9}$$

where F_l is the set of ground formulas with weight w_i that X_l appears in, and $f_i(X_l = x_l, B_l = b_l)$ is the value(0 or 1) of the feature corresponding to the ith ground formula when $X_l = x_l$ and $B_l = b_l$; $P(X_l = x'|B_l = b_l)$ is the probability that $X_l = x'$ holds given Markov blanket b_l.

In such a way, we can use this gradient-descent method to update the weight of each formula in MLN in each state transition process. Different from MLN, we introduce state and action, which only exist in RL, to MLN. The original weight learning methods in MLN use an evidence database to do gradient-descent learning while in RL we have states, actions and state transitions which can't be represented simply in an evidence database. That's why we can't directly use the weight learning methods of MLN. Now we can see that MLN helps RL to choose actions while RL helps MLN to learn weights. We call this method reinforcement learning algorithm with Markov logic networks.

4.2 RLMLN with Eligibility Trace

Combining with eligibility trace, we can get the backward view([7]) of our RLMLN algorithm.

$$Q^{\pi}(s,a) = \ln((r + \gamma \max_{a'} P(A = a'|S = s')))$$
$$Q(s,a) = \ln P(A = a|S = s)$$
$$\delta = Q^{\pi}(s,a) - Q(s,a)$$
$$\vec{w}_{t+1} = \vec{w}_t + \alpha\delta \frac{\partial}{\partial w_t} Q(s,a)\vec{e}_t$$
$$\vec{e}_t = \gamma\lambda\vec{e}_{t-1} + \frac{\partial}{\partial w_t} Q(s,a)$$
$$\frac{\partial}{\partial w_i} Q(s,a) = \frac{\partial}{\partial w_i} \ln P(A = a|S = s)$$
$$\frac{\partial}{\partial w_i} \ln P(X_l = x_l|B_l = b_l)$$
$$= \sum_{f_i \in F_l} \left\{ f_i(X_l = x_l, B_l = b_l) - \sum_{x' \in 0,1} f_i(X_l = x', B_l = b_l) P(X_l = x'|B_l = b_l) \right\}$$

where $Q^{\pi}(s,a)$ is the exact, correct value for state-action pair (s,a), and here we use an estimate $\ln((r + \gamma \max_{a'} P(A = a'|S = s')))$ to substitute it. $\vec{w}_t$ is the weights of formulas in MLN; $Q(s,a)$ is the Q-value of state-action pair (s,a) given by MLN inference.

A description of this algorithm is given in Fig. 4. In this algorithm, we get $P(a|s)$ from the inference of MLN(Equ. 5) and then apply RL to update the weight of each formula(Equ. 8). Our implementation is based on the open source project–Alchemy.

5 Experiment on Blocks World

5.1 Blocks World Problem

A blocks world contains a fixed set of blocks. Each block has two positions:on the top of another block or on the floor. The action in this world is `move(a, b)`, where `a` must be a block and `b` is either a block or the floor. Marina Irodova and Robert H. Sloan([11]) applied RL and function approximation to this task and got a better result than RRL. However, at present it is still a difficult task in AI. In our experiment, we also consider three goals as other researchers do.

Stack States with all blocks in a single stack
Unstack States with every block on the floor
On(a, b) States with block `a` on top of block `b`

For ease of introducing prior knowledge, we introduce three predicates to blocks world, they are `ontop(x, y)`, `above(x, y)`, `maxheight(x)`. Predicate

Initialize $\vec{w}$ of MLN formulas with prior knowledge or just a constant arbitrarily chosen, $\vec{e} = \vec{0}$
Repeat(for each episode):
  $s \leftarrow$ initial state of episode
  For all $a \in \mathcal{A}(s)$:
    $Q_a \leftarrow P(A = a|S = s)$ //(MLN)
  $a \leftarrow \arg\max_a Q_a$
  With probability $\epsilon : a \leftarrow$ a random action $\in \mathcal{A}(s)$
  Repeat(for each step of episode):
    Take action a, observe reward, r, and next state s'
    $Q(s, a) = \ln Q_a$
    For all $a \in \mathcal{A}(s')$
      $Q_a \leftarrow P(A = a|S = s')$ //(MLN)
    $a' \leftarrow \arg\max_a Q_a$
    With probability $\epsilon : a' \leftarrow$ a random action $\in \mathcal{A}(s')$
    $Q^{\pi}(s, a) = \ln(r + \gamma Q_{a'})$
    $\delta \leftarrow Q^{\pi}(s, a) - Q(s, a)$
    $\frac{\partial}{\partial w_i} Q(s, a) = \frac{\partial}{\partial w_i} \ln P(A = a|S = s)$ //(MLN, Equ. 9)
    $\vec{w}_{t+1} = \vec{w}_t + \alpha\delta \frac{\partial}{\partial w_t} Q(s, a)\vec{e}_t$
    $\vec{e}_{t+1} = \gamma\lambda\vec{e}_t + \frac{\partial}{\partial w_t} Q(s, a)$
    $a \leftarrow a'$
  until s' is terminal

Fig. 4. Reinforcement learning algorithm with Markov logic networks

`ontop(x, y)` means block `x` is on the top of block `y`; `above(x, y)` means block `x` is above block `y`; and `maxheight(x)` means block `x` has the maximum height than other blocks. We define height of block `x` like this: if `on(x, Floor)`, then `height`$(x) = 0$, and if `on(x, y)`, `height`$(x) =$ `height`$(y) + 1$. Predicates `ontop(x, y)`, `above(x, y)` can be defined using `on(x, y)`, `clear(x, y)` recursively as Markov nets allow cycles.

{`on(A, B)`, `on(B, Floor)`, `on(E, C)`, `on(E, C)`, `on(C, D)`, `on(D, Floor)`} is the initial state in our experiment. The predicate to be inferred is `move`. In our experiment, the goal state for *on* task is {`on(B, D)`}. The goal state of *unstack* and *stack* task is trivial. MLN formulas for the three tasks are given below:
On task:

```
//predicates
on(bl, bl)
move(bl, bl)
clear(bl)
maxheight(bl)
ontop(bl, bl)
above(bl, bl)
//formulas
1.0 clear(B) ^ clear(D) => move(B, D)
```

```
1.0 ontop(x, B) => move(x, Floor)
1.0 ontop(x, D) => move(x, Floor)
1.0 on(x, y) => above(x, y)
1.0 on(x, y) ^ above(y, z) => above(x, z)
1.0 clear(y) ^ above(y, x) => ontop(y, x)
```

Stack task:

```
1.0 maxheight(x) ^ clear(y) => move(y, x)
```

Unstack task:

```
1.0 !on(a1,Floor) ^ clear(a1) => move(a1,Floor)
```

In the stack and unstack MLN formulas, we omit the predicate declarations so as to avoid duplication. At the rest of this paper we also omit the declarations in MLN formulas.

5.2 Experimental Results

In this section, we perform an experiment to validate our RLMLN algorithm. The problem domain is a 5-block blocks world task as shown in Fig.5. Fig. 6 gives the experiment result. Parameter setting in our experiment is $\alpha = 0.1, \gamma = 0.99$ and $\epsilon = 0$. Taking account of the accuracy of inference and equivalent initial weight of MLN formulas , we choose actions randomly among the actions whose probability is near the maximum probability, that's why we set $\epsilon = 0$ in this task. Another reason is that we have provided very exact prior knowledge for every task and a positive ϵ will cause unnecessary fluctuation. For different task(*stack, unstack, on*), reward is 1 when the goal is reached, 0 otherwise. No comparisons are provided here because the learned results are optimal for the three tasks and as a result comparisons are not necessary on this problem. Further study of RLMLN on other domains, like Tetris, is undertaken and comparisons on these domains will be taken in the near future.

5.3 Transfer to Larger Blocks World Problems

For blocks world task, we find that our formulas above can easily transfer to larger blocks world problems. We design an experiment of a *on* task with 10 blocks. The task starts with a random initial state and the goal state is $\{\texttt{on(B7,B3)}\}$. We use the above learned MLN formulas with only `B`, `D` changed to `B7`, `B3`. We run the experiment twice. The result is given in Fig. 7 and the initial states are(randomly generated in one run):
S1:

$$\{\texttt{on(B9,Floor)}, \texttt{on(B8,B7)}, \texttt{on(B7,B6)}, \texttt{on(B6,Floor)}, \texttt{on(B5,B4)}, \texttt{on(B4,B3)}, \texttt{on(B3,Floor)}, \texttt{on(B2,B1)}, \texttt{on(B1,B0)}, \texttt{on(B0,Floor)}\}$$

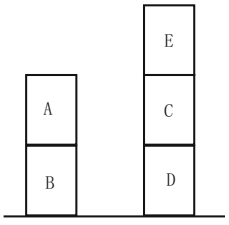


Fig. 5. Stack task

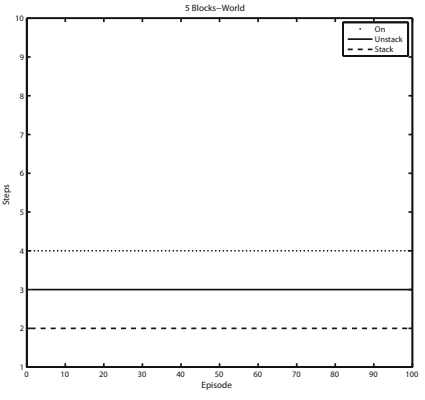


Fig. 6. Experiment with 5 blocks

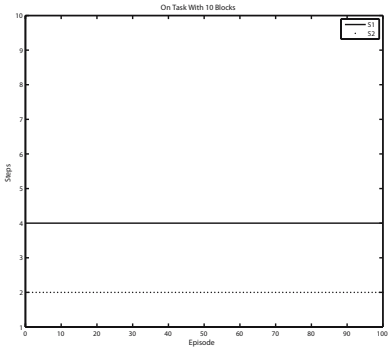


Fig. 7. Transfer to 10 Blocks On Task

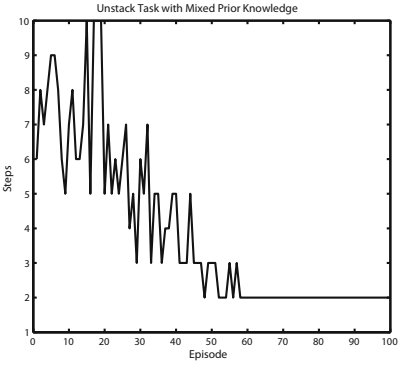


Fig. 8. Experiment with Mixed Knowledge

S2:

$\{\texttt{on}(\texttt{B0},\texttt{Floor}), \texttt{on}(\texttt{B1},\texttt{B0}), \texttt{on}(\texttt{B2},\texttt{B1}), \texttt{on}(\texttt{B3},\texttt{B2}), \texttt{on}(\texttt{B4},\texttt{Floor}), \texttt{on}(\texttt{B5},\texttt{B4}),$
$\texttt{on}(\texttt{B6},\texttt{B5}), \texttt{on}(\texttt{B7},\texttt{B6}), \texttt{on}(\texttt{B8},\texttt{B7}), \texttt{on}(\texttt{B9},\texttt{Floor})\}$

5.4 Mix the Prior Knowledge

In the above experiment, we give an exact prior knowledge about blocks world, so the learning process seems to become useless. The reason is that blocks world problem is easy for humans and we can easily figure out the exact prior knowledge. However, for complex problems like Tetris, it's difficult to give an exact prior knowledge and then we will see the power of RLMLN to distinguish good

formulas and bad formulas(by modifying the weights). In this subsection, we put all the formulas for the three different tasks and do an experiment on unstack task with 4 blocks to validate the power of RLMLN in adjusting weights or selecting formulas(or prior knowledge). In order to confuse RLMLN at the beginning, we set the formulas that are against the unstack goal with higher weights. The initial MLN is:

```
2.0 clear(B2) ^ clear(B0) => move(B2,B0)
1.0 ontop(x,B2) => move(x,Floor)
1.0 ontop(x,B0) => move(x,Floor)
1.0 on(x,y) => above(x,y)
1.0 on(x,y) ^ above(y,z) => above(x,z)
1.0 clear(y) ^ above(y,x) => ontop(y,x)
1.0 clear(a1) ^ !on(a1,Floor) => move(a1,Floor)
2.0 maxheight(x) ^ clear(y) => move(y,x)
```

The learned formulas are given below:

```
1.99943  move(B2,B0) v !clear(B2) v !clear(B0)
1.47049  !ontop(a1,B2) v move(a1,Floor)
1.47049  !ontop(a1,B0) v move(a1,Floor)
1        !on(a1,a2) v above(a1,a2)
1        !on(a1,a2) v above(a1,a3) v !above(a2,a3)
1        ontop(a1,a2) v !above(a1,a2) v !clear(a1)
1.47049  on(a1,Floor) v move(a1,Floor) v !clear(a1)
1.96995  move(a1,a2) v !maxheight(a2) v !clear(a1)
```

In this experiment, $\epsilon = 0.1$ and the other parameters are the same as those used above. We can see the weights of the formulas fit the task are incremented in the learning process while the others are decreased or remain. We run the experiments 10 times and the initial state is generated from a random number generator with the same seed $\texttt{seed} = \texttt{time}(\texttt{NULL})$ in the 10 runs. The result is shown in Fig. 8.

From the experiment result, we can see that our new method works very well on blocks world problem when we introduce some prior knowledge. If the formulas are abstract and exact enough, the learned MLN formulas can be easily transfered to larger tasks. On other tasks other than blocks world, the learned MLN may need some modification or relearning from the learned formulas. Of course we can use function approximation on this problem, however, finding good features is still a difficult and arduous job. With RLMLN, finding a solution for such tasks becomes easier and faster as the formulas give us a high level and comprehensible abstraction of this problem while function approximation in RL often introduces obscure features and needs some programming tricks. Currently, we are taking a study on applying RLMLN to Tetris and in the near future we will be able to give a better example of RLMLN's applications.

6 Conclusion and Future Work

In this paper, we propose a reinforcement learning algorithm with Markov logic networks (RLMLN). This method helps us represent states in RL compactly and introduce prior knowledge to a learning system. Furthermore, figuring out a set of formulas of RLMLN is easier than finding features for function approximation in some problem domains, such as the blocks world. Experimental results on blocks world show that RLMLN is a promising method. If the problem can be formulated in RRL framework, like blocks world here, the learned RLMLN can be easily transfered or directly used to larger tasks. However, not all tasks can easily be formulated as relational problems, which inludes mountain car and acrobot, etc([12]).

We are now planning to extensively examine the performance of RLMLN on more problem domains, like Tetris. However, hand coded formulas may be not enough for such a task, so structure learning of RLMLN will be also a collar work for us in the near future.

A recursive random field(RRF)([13]) is a representation where MLNs can have nested MLNs as features. Recursive random fields(RRFs) have more power than MLNs as they can represent disjunctions and conjunctions ,and universal and existential quantifiers symmetrically. As a result, we are now also considering to upgrade MLN in RLMLN to RRF.

Acknowledgements

The work is supported by the Natural Science Foundation of China (No.60775046 and No.60721002). We would like to thank Yinghuan Shi and Tianyin Xu for helpful comments on the preliminary draft.

References

1. van Otterlo, M.: A Survey of Reinforcement Learning in Relational Domains. Technical Report TR-CTIT-05-31, University of Twente, CTIT Technical Report Series, ISSN 1381-3625 (July 2005)
2. Kersting, K., De Raedt, L.: Logical Markov Decision Programs. In: Proceedings of the IJCAI 2003 Workshop on Learning Statistical Models of Relational Data (2003)
3. Richardson, M., Domingos, P.: Markov Logic Networks. Machine Learning (2006)
4. Singla, P., Domingos, P.: Entity Resolution with Markov Logic. In: IEEE International Conference on Data Mining (2006)
5. Mihalkova, L., Mooney, R.J.: Transfer Learning with Markov Logic Networks. In: Proceedings of the ICML Workshop on Strutural Knowledge Transfer for Machine Learning (2006)
6. de Raedt, L., Driessens, K.: Relational Reinforcement Learning. Machine Learning 43, 7–52 (2001)
7. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)

8. Domingos, P., Richardson, M.: Markov Logic: A Unifying Framework for Statistical Relational Learning. In: ICML 2004 Workshop on Statistical Relational Learning (2004)
9. Domingos, P., Richardson, M.: Unifying Logical and Statistical AI. In: Proceedings of AAAI (2006)
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
11. Irodova, M., Sloan, R.H.: Reinforcement Learning and Function Approximation. In: The Florida AI Research Society Conference (2005)
12. Taylor, M.E., Kuhlmann, G., Stone, P.: Autonomous Transfer for Reinforcement Learning. In: Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2008) (2008)
13. Lowd, D., Domingos, P.: Recursive Random Fields. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007) (2007)

Learning the Filling Policy of a Biodegradation Process by Fuzzy Actor–Critic Learning Methodology

Efraín Franco Flores[1,⋆], Julio Waissman Vilanova[1,2], and Jair García Lamont[1]

[1] Centro de Investigación en Tecnologías de la Información y Sistemas
Unversidad Autónoma del Estado de Hidalgo
Carr. Pachuca-Tulancingo Km. 4.5, Pachuca Hgo. 42084, México
[2] Departamento de Matemáticas Universidad de Sonora Blvd. Encinas y Rosales,
Hermosillo Son. 83000, México
rick_honter@hotmail.com, juliowaissman@mat.uson.mx,
nneurotronica@hotmail.com

Abstract. Reinforcement Learning is a learning methodology through an agent interacting with the environment. Actor-Critic methods have a separated memory structure to present the independence between the policy and the value function. In systems where the states are defined as continuous, there is a problem with dimensionality, and an approximation method has to be used. The classic Reinforcement Learning algorithms can be combined with Fuzzy Logic techniques to store all value functions, since Fuzzy Logic has been proved to be an effective universal approximator. This work propose a Fuzzy Actor-Critic method to compute and store the state values using fuzzy logic to get a state approximation. Phenol is one of the most important water pollutants on chemistry industries. A phenol biodegradation process consist on a Sequence Batch Reactor (SBR), that need an near optimal filling policy for its correct operation. Fuzzy Actor Critic learning strategy offers a operation policy and it can be linguistically interpreted by the process experts, this approach can be useful to propose a comprehensive filling policy of a biodegradation SBR process.

1 Introduction

Reinforcement Learning, RL [1] is a methodology that needs a goal to achieve an objective. The way of learning in through an agent and an environment. The principal idea of this technique consists on a agent interacting with the environment.

It consists on an agent who takes actions that modify the environment by changing from one state to another state, until a final state is visited. The action a, determines the next visited state. There are two states, one at the present time t, named s_t, and the next, that comes by applying an action at the time

⋆ PROMEP Scholarship Master Degree Program in Computer Science.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 243–253, 2008.

t, denoted by s_{t+1}. The convenience of being in a state is named *reward*, r, and it is a numeric value assigned to every change. The form of *how good* an agent *learns* is in function of the total reward that this have acquired.

There are 4 principal subelements in a Reinforcement Learning system, a *policy*, π, that defines the behavior form of the agent; a *reward function*, r_t, that assigns to the ambient's state action pair a number, and the agent's objective will be to maximize the total reward who receives during the entire process; a *value function*, $V^{\pi}(s)$, that indicates the convenience of being in that state; and the *model* of the environment, that mimics the system behavior. The actualization rule of the *value function* is defined by the Temporal Difference methods [2].

In models where the states are defined as continuous, there is a problem with dimensionality, and an approximation method has to be used. The classic Reinforcement Learning algorithms can be combined with Fuzzy Logic techniques to store the values of V, since Fuzzy Logic has been proved to be an effective universal approximator on problems with continuous values [3].

Phenol is one of the most important intermediate on chemistry industries. It is considered as a common and dangerous contaminant on hydric environments, contained in residual water of some chemistry industries, oil refineries and some other companies. That's why several studies have been made in order to develop efficient decontamination methods and economically viable for phenol degradation [4].

One of these methodologies consist on a Sequence Batch Reactor *(SBR)*, that contains a group of microorganism that degrade phenol [5]. This SBR need an optimal policy to its operation. To find this optimal policy an actor-critic method can be applied to find it.

A first approach to combine the reinforcement learning with fuzzy logic was presented by [6] using the Q–learning algorithm. Duan and Xu use the Fuzzy Q-Learning algorithm applied to a robot navigation [7]. In [8] a Fuzzy Actor Critic algorithm is proposed and extended in [9]. This approach was applied to power control in wireless transmitters. A neuro–fuzzy Actor-Critic reinforcement learning network was developed in [10], and used in the classical cart–pole problem. In these works, the reinforced signal is used in order to learn the fuzzy and/or neural parameters of the fuzzy inference system.

In this work, the fuzzy inference system are used to represent the state of the environment in a linguistically interpretable way to the process experts and the reinforcement learning is applied to select the filling policy. The paper is organized has follows: section 2 describes the classic actor-critic method; in section 3 the fuzzy actor-critic algorithm is developed; section 4 presents the case of study, a SBR for phenol degradation model; in section 5 the simulation results are discussed and, finally, conclusions are on section 6.

2 Actor - Critic Methods

Actor-Critic methods have a separated memory structure to present the independence between the policy and the value function. The policy structure, known as

the *Actor*, is used for selecting actions. The estimated value function is named the *Critic*, because it evaluates the actions taken by the actor [11]. The critic learns and evaluates any policy followed by the actor.

The critic is a state-value function. After every action selection, the critic evaluates the new state to determine if the agent is on a better or worst moment than the expected by

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t). \tag{1}$$

The fortification or attenuation can be implemented increasing the probability of being in the state s_t applying the action a_t, denoted as $p(s_t, a_t)$, with

$$p(s_t, a_t) \leftarrow p(s_t, a_t) + \beta\delta_t, \tag{2}$$

where β is positive step-size parameter. The methodology of this learning is showed on the algorithm 1.

```
1  Initialize V(s) and p(s,a) arbitrarily
2  Repeat
3  |  Initialize s
4  |  Repeat
5  |  |  Select a' from s' using the policy derived from V.
6  |  |  Take the action a, observe r,s'.
7  |  |  δ_t ← r + γV(s') − V(s).
8  |  |  p(s,a) ← p(s,a) + βδ_t.
9  |  |  V(s) ← V(s) + αδ_t.
10 |  |  s ← s'.
11 |  Until s is terminal ;
12 Until the final episode ;
```

Algorithm 1. Actor - Critic

3 Fuzzy Actor-Critic

The classic Actor-Critic methods are good algorithms to estimate a policy of a discrete system. When an environment is described by a continuous state space, there are some difficulties to store the value function information, because the states number turns to infinite. This article propose a Fuzzy Actor-Critic method to compute and store the state values using fuzzy logic to get a state approximation.

The state function value will be $V(s)$ and it will be computed from a *Critic* fuzzy inference system. The estimation of the actions by rule is realized with the probability matrix P, with entries $p(i, j)$, where i is the fuzzy rule and j is one of the possible actions in the state s. Then, the action $a(i, j)$ is the action to be taken in the state s for rule i, with a probability $p(i, j)$.

So, for the *Actor*, the action $a(s)$ chosen at state s is obtained by a fuzzy inference system, where the i^{th} rule is described as follows:

$$\begin{aligned} \textbf{if} \;\; s \text{ is } S_i \; \textbf{then} \;\; & a(i,1) \text{ with } p(i,1) \\ \textbf{or} \;\; & a(i,2) \text{ with } p(i,2) \\ & \cdots \\ \textbf{or} \;\; & a[i,J] \text{ with } p[i,J], \end{aligned}$$

where J is the total number of actions and S_i is the linguistic label that describes the antecedent of the rule.

If the learning was already done, a *greedy* policy is implemented and, the action chosen for this rule, a_i, will be the action $a(i,j)$ with the bigger $p(i,j)$. In a learning stage, the compromise of exploitation and exploration of the state space is crucial for the reinforcement learning methodology. So, in the learning stage a *softmax* [12] policy will be used.

The estimation of the consequent global action is realized with the COA defuzzification method

$$a(s) = \frac{\sum_{i=1}^{N} \mu_i(s) \times a_i}{\sum_{i=1}^{N} \mu_i(s)}, \tag{3}$$

where $\mu_i(s)$ represents the membership of the state s into the rule i, "s is S_i", and N is the total number of rules.

For the *Critic*, a fuzzy inference system with the same antecedents structure of the *Actor* ones is used, *i.e.*, for the rule i

$$\textbf{if} \;\; s \text{ is } S_i \; \textbf{then} \; v_i$$

where v_i is the state value for the rule i. Then, the state value for the state s is obtained with

$$V(s) = \frac{\sum_{i=1}^{N} \mu_i(s) \times v_i}{\sum_{i=1}^{N} \mu_i(s)}. \tag{4}$$

For learning, the estimated global action $a(s)$ is applied to the environment, and the reward r and the next state s' are obtained. With the new state, s', the value function $V(s')$ is computed with (4).

The error signal estimation will be given by

$$\Delta V = r + \gamma V(s') - V(s), \tag{5}$$

where $0 < \gamma < 1$ is the discount rate. The actualization of the probability matrix will be calculated as it follows:

$$p(i, a(s)) = p(i, a(s)) + \epsilon_1 \times \Delta V \frac{\mu_i(s)}{\sum_{j=1}^{N} \mu_j(s)}. \tag{6}$$

where $0 < \epsilon_1 < 1$ is a learning factor. From equation (6) is possible to see that the probability of the chosen action is actualized, in each rule, proportionally to the participation degree of that rule in the global decision.

Then, the state value v_i from each rule is actualized as

$$v_i = v_i + \epsilon_2 \times \Delta V \frac{\mu_i(s)}{\sum_{j=1}^{N} \mu_j(s)}. \tag{7}$$

where $0 < \epsilon_2 < 1$ is a learning factor. Typically, the factors γ and ϵ_1 are values near to 1, and the factor ϵ_2 is near to $1/k$, where k is the number of possible actions.

The *Fuzzy Actor-Critic* algorithm is showed on the algorithm 2.

1 Initialize a Fuzzy Inference System with $\mu_i(s)$ the membership function for rule i.
2 Initialize $p(s_i, a)$, v_i arbitrarily for all rule and for all a
3 **Repeat**
4 Initialize s
5 **Repeat**
6 **for** *each rule i* **do**
7 Compute the membership functions $\mu_i(s)$.
8 Select a_i using a policy (*that is* Softmax).
9 **end**
10 $a(s) \leftarrow \frac{\sum_{i=1}^{N} \mu_i(s) \times a_i}{\sum_{i=1}^{N} \mu_i(s)}$.
11 $V(s) \leftarrow \frac{\sum_{i=1}^{N} \mu_i(s) \times v_i}{\sum_{i=1}^{N} \mu_i(s)}$.
12 Apply $a(s)$ and observe the new state s' and reward r.
13 $V(s') \leftarrow \frac{\sum_{i=1}^{N} \mu_i(s') \times v_i}{\sum_{i=1}^{N} \mu_i(s')}$.
14 $\Delta V \leftarrow r + \gamma V(s') - V(s)$.
15 **for** *each rule i* **do**
16 $p(i, a_i) \leftarrow p(i, a(s)) + \epsilon_1 \times \Delta V \frac{\mu_i(s)}{\sum_{j=1}^{N} \mu_j(s)}$.
17 $v_i \leftarrow v_i + \epsilon_2 \times \Delta V \frac{\mu_i(s)}{\sum_{j=1}^{N} \mu_j(s)}$.
18 **end**
19 $s \leftarrow s'$.
20 **Until** *s is terminal* ;
21 **Until** *the final episode* ;

Algorithm 2. Fuzzy Actor-Critic

4 *SBR* Phenol Biodegradation

SBR belong to the activated sludge technologies because it retains a high cell density inside it. It is constituted on a recipient filled with residual water during a finite time $T > 0$, and then it works as a discontinued reactor. After the required treatment, the suspension is decanted and the remainder is taken from the recipient. A SBR cycle consists on five phases: filled, reaction, decantation, emptying, and a time out. The phase time is estimated by the characteristics of the residual water on treatment and the effluent purification requirements.

The SBR presents some advantages with respect to the continuous activated sludge process. Since SBR is a batch process, the effluent can be held in the reactor until it is treated if there is somewhere far the influent to be stored. This can minimize the deterioration of effluent quality associated with influent spikes. Also, biomass will not be washed out of an SBR because of flow surges. In addition, settling occurs when there is no inflow or outflow. However, the SBR systems have also some disadvantages, generally related to a higher level of control sophistication.

Real-time control of SBR processes still represent a challenging area of endeavor. In particular, the control design is difficult principally because the processes involving living microorganisms exhibit large nonlinearities, strongly coupled variables and often poorly understood dynamics. The control of the SBR process is performed in the filling phase. During the reactor filling, contaminated water with xenobiotic material to degrade is deposited in it. The filling process can be done in two ways.

First, following an *a priori* filling policy in order to maximize the degradation of the substrate. In order to propose a filling profile, it is important that the policy maximise the phenol consumption and, at least, assure that at the end of the working cycle all the pollutant metabolic intermediates are totally consumed. In wastewater processes, a error in the operation can damage the biological process for several week, even months. That is the reason why the policy has to be interpretable by the process operators. Also, a restrictions in the actuator (valves or pumps, normally) has to be considered. The other strategy is to develop an adaptive feedback control, but, normally the real-time monitoring of many key process variables, which are needed by advanced control algorithms, is hampered by the lack of reliable on-line sensors.

Like Fuzzy Actor Critic learning strategy offers a operation policy and, in addition, this policy can be linguistically interpretable by the process experts, this approach can be useful to propose a comprehensive filling policy of the reactor. The goal is to generate a filling policy that avoid the intermediate remainder at the end of the reaction phase, and optimize the phenol consumption.

A model of a phenol degradation in a SBR, with a inhibitory was developed and experimentally validated in [5]. The model is expressed by the equations:

$$\frac{dX_1(t)}{dt} = \mu_1(t)X_1(t) - \frac{F_{in}(t)}{V(t)}X_1(t), \tag{8}$$

$$\frac{dX_2(t)}{dt} = \mu_2(t)X_2(t) - \frac{F_{in}(t)}{V(t)}X_2(t), \tag{9}$$

$$\frac{dS_1(t)}{dt} = -\frac{\mu_1(t)}{Y_1}X_1(t) + \frac{F_{in}(t)}{V(t)}(S_1^* - S_1(t)), \tag{10}$$

$$\frac{dS_2(t)}{dt} = \alpha\mu_1(t)X(t) - \frac{\mu_2(t)}{Y_2}X_2(t) - \frac{F_{in}(t)}{V(t)}S_2(t)), \tag{11}$$

$$\frac{dV(t)}{dt} = F_{in}(t) \tag{12}$$

Table 1. Model's parameters

Parameter	Value	Units
$\mu_{\max_1}$	0.39	1/h
Y_1	0.57	mg/mg
K_{S_1}	30	mg/l
K_{i_1}	170	mg/l
K_2	160	mg/l
$\mu_{\max_2}$	0.028	1/h
Y_2	0.67	mg/mg
K_{S_2}	350	mg/l
K_1	66	mg/l
α	1.6	mg/l

where $S_1(t)$ is the phenol concentration, $S_2(t)$ is the metabolic intermediate concentration, $X_1(t)$ is the biomass concentration that consume phenol and produce metabolic intermediate, $X_2(t)$ is the biomass concentration that consume intermediate, $F_{in}(t)$ is the influent stream y $V(t)$ is volume. The specific growth rates are defined by $\mu_1(t)$ and $\mu_2(t)$, that is the specific biomass growth rate; Y_1 and Y_2 are, respectively , the constant yields for phenol and intermediate consumption; α is the production constant of the intermediate. And, S_1^* is the phenol concentration on the influent, which is constant.

The specific microbian growths rate is calculated by:

$$\mu_1(t) = \frac{\mu_{\max_1} S_1(t)}{K_{S_1} + S_1 + S_1^2/K_{i_1}}, \tag{13}$$

$$\mu_2(t) = \frac{\mu_{\max_2} S_2(t)}{K_{S_2} + S_2(t) + S_2^2/K_{i_2}}. \tag{14}$$

where μ_{max_1} is the growth maximum value due to phenol concentration, μ_{max_2} is the maximum value due to the intermediate concentration. The parameters K_{S_1} and K_{S_2} are the limitation parameters and K_{i_1} and K_{i_2} are the inhibition constants.

The retained parameter values have been validated and reported by [5] based on an experimental work and a sensibility model analysis. The initial values used for the simulation are given by $X(0) = 1000$ mg/l, $S_1(0) = 200$ mg/l, $S_2(0) = 0$ mg/l and $V(0) = 1$l. The reaction phase time is $15hr$.

5 Simulation Results

The proposed Fuzzy Actor Critic Algorithm was applied to the SBR process. For modelling the Fuzzy Inference System three variables were used has antecedents: Phenol concentration, the intermediate and the operation time.

The goal of the optimization algorithm is to maximize the phenol consumption in a SBR working cycle, assuring that the phenol and the metabolic intermediate

are practically consumed at the end of the reaction phase. The actions applied at the beginning of the process are different as the ending process actions, due to the phenol consumption needs to be efficient at the end. So, is very important that the variable of time is considered, to assure that the actions at the beginning will produce biomass and the actions at the end degrade phenol quickly each time cycle.

Each variable uses a particulary membership function combination, in order to reduce the rule dimensionality size. The objective of this combination is to accelerate and simplify the entire learning process. The membership functions per variable are showed in the figure (1). The parameters taken by the other membership functions were proposed by the SBR process expert, looking to empathize the membership limits of each variable to degrade phenol and intermediate in an optimal way. Two membership functions were defined for the time variable, the membership function for *Reaction phase* is a *L*–type with parameters in 11 h and 13 h. The membership function for *End of working cycle* is defined has the complement of the membership function of *Reaction phase*.

A mixed reward's strategy was implemented. In the reaction phase, the reward is assumed like the phenol feeding at this instant. This positive reinforcement works in a periodic (batch) like learning operation.

At the end of reaction phase, the reward is defined as -1000 if phenol concentration is up to 10 mg/l (a high penalization), and -100 if intermediate is up to 10 mg/l (a lower penalization). This reinforced signal is applied in a continuous–like learning method. This penalization is taken in account in the next SBR simulated working cycle.

A discreet control policy with a sampling period of 0.1 hours was used. This sampling time is large enough to the existing on line phenol sensors and is

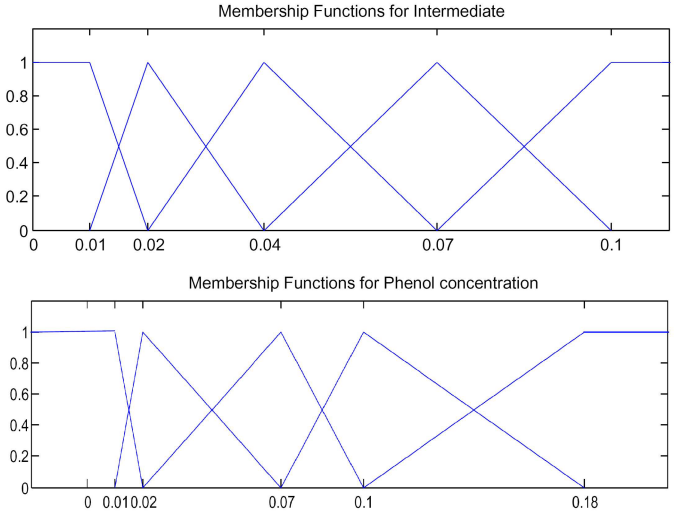


Fig. 1. Membership Functions defined by the expert

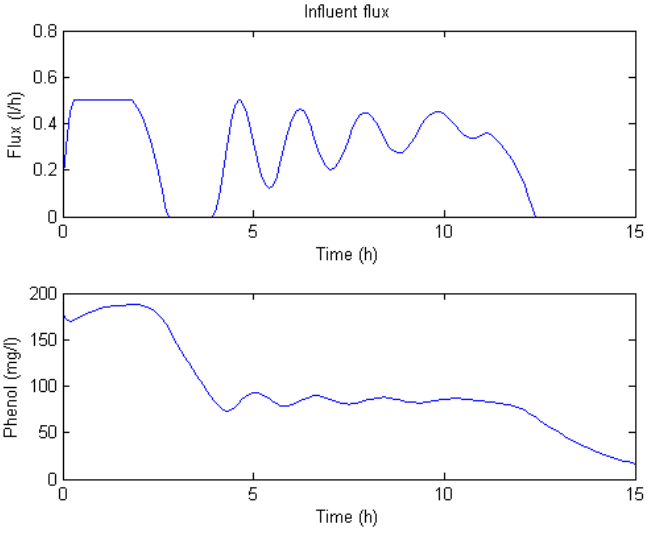


Fig. 2. The Phenol degradation at the end of the last three cycles

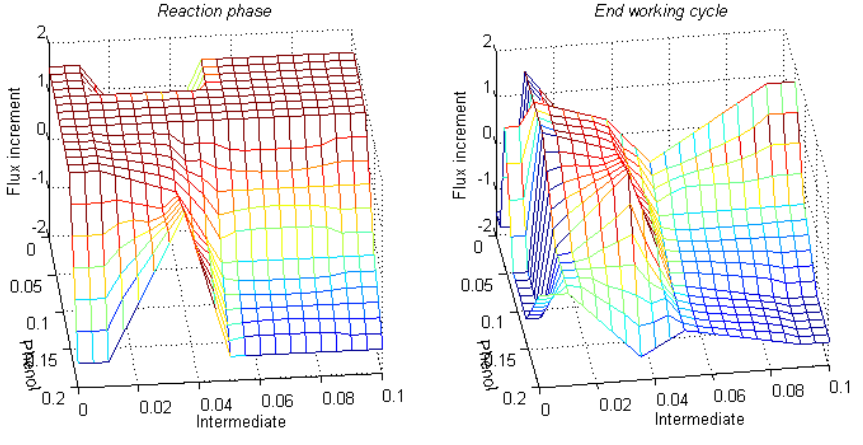


Fig. 3. Response surface of the fuzzy policy for the two linguistic values of t

smaller than the dynamics of the process. The influent phenol concentration was established at 500 mg/l and the control of the filling profile will be controlled by the increment of influent flux (l/h). The possible action was the same for every possible state $\Delta a \in \{-1.5, -1, -0.5, 0, 0.5, 1, 1.5\}$ multiplied by the sampling time (for numerical integration). The learning parameters were:$\lambda = 0.9$, $\epsilon_1 = 0.05$, $\epsilon_2 = 0.9$.

In figure (2) the filling policy is showed with the phenol profile. The phenol consumption in the near optimal policy was assuring a total phenol and metabolic

consumption at the end of the cycle. A oscillations in the phenol profile is present manly due to the low resolution of the actuator (a *masterflex* filling pump).

In figure (3) the actions taken for the possible values of phenol and metabolic intermediate are shown. In the left hand, the operation policy for reaction phase are displayed. At low and very hight concentration of phenol and medium concentration of intermediate, the learning is not god enough. That is because these states are not commons for the system operation. In the right hand, a response surface for the end of working cycle is presented. The learning of the ending phase is more complicate because it depends on the batch learning rewards.

6 Conclusions

In this paper a Reinforcement Learning Algorithm based on Actor Critic extended to Fuzzy Inference Systems has been developed. FIS learn better when a short rule number is used, so, the execution time and learning is simplified.

The developed algorithm was applied to a biotechnological phenol degradation process. For the learning, a mixed schema (batch and continuous) was used to penalize the residual phenol at the end of the work cycle so as the remaining metabolic intermediate, while the higher phenol consumption was rewarded during the reaction phase of the SBR process. A filling reactor policy was obtained, this, to improve the SBR efficiency at simulation level.

This policy has a meaning to the process operators. It is necessary to carry out experiments into the pilot plant in order to validate experimentally the policy.

The learning method depends of a good FIS selection, at the same time that this has to store the linguistic meaning to the operators process. That's why the classic FIS generation techniques (*i.e.* Fuzzy C-Means, Subtractive Clustering) can not be used directly. An adjustment parameter method without loosing the linguistical meaning of membership functions is under development.

References

1. Bertsekas, D., Tsitsiklis, J.: Neuro-Dynamic Programming. Athena Scientific, Belmont (1996)
2. Sutton, R.: Learning to predict by the method of temporal differences. Machine Learning 3, 9–44 (1988)
3. Castro, J.: Fuzzy logic controllers are universal aproximators. IEEE Trans. on Systems, Man and Cybernetics 25, 629–635 (1995)
4. García, J.A., Ramírez, C., Ben-Youssef, C., Waissman, J., Vázquez, G.A.: Modelado de la biodegradación de fenol por lodos activados aclimatados. Revista Internacional de Contaminación Ambiental 21, 802–807 (2004)
5. Vázquez, G., Ben Youssef, C., Waissman, J.: Two-step modeling of the biodegradation of phenol by an acclimated activated sludge. Chemical Engineering Journal 117, 245–252 (2006)
6. Glorennec, P.Y., Jouffe, G.: Fuzzy Q–learning. In: Proceedings of Fuzz-IEEE, Sixth International Conference on Fuzzy Systems, Barcelona, pp. 659–662 (1997)

7. Duan, Y., Xu, X.: Fuzzy reinforcement learning and its application in robot navigation. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, pp. 899–904 (2005)
8. Vengerov, E., Berenj, H.: A convergent actor–critic-based FRL algorithm with application to power management of wireless transmitters. IEEE Transactions on Fuzzy Systems 11, 478–485 (2003)
9. Vengerov, E., Bambos, N., Berenj, H.: A fuzzy reinforcement learning approach to power control in wireless transmitters. IEEE Trans. on Systems, Man and Cybernetics 35, 765–778 (2005)
10. Wang, X., Cheng, Y., Yi, J.: A fuzzy actor–critic reinforcement learning network. Information Sciences 178, 3764–3781 (2007)
11. Barto, A., Sutton, R.S., Anderson, C.W.: Neuronlike elements that can solve difficult learning control problems. IEEE Trans. on Systems, Man and Cybernetics 13, 1038–1044 (1983)
12. Sutton, R., Barto, A.: Reinforcement Learning, an Introduction. MIT Press, Cambridge (1998)

INDIE: An Artificial Immune Network for On-Line Density Estimation

Juan Carlos Galeano-Huertas and Fabio A. González

Intelligent Systems Research Lab.
National University of Colombia
{jcgaleanoh,fagonzalezo}@unal.edu.co

Abstract. This paper presents a new artificial immune network model that addresses the problem of non-parametric density estimation. The model combines immune ideas with the known Parzen window estimator. The model uses a general representation of antibodies, which leads to redefine the network dynamics. The model is able to perform on-line learning, that is to say, training samples are presented only once. Results from exploratory experiments are presented in order to give insights on the reliability of the estimations of the proposed model.

1 Introduction

Density estimation is an important problem in both statistics and machine learning fields, since it represents a statistical approach for basic learning tasks, such as classification [1]. The problem can be stated as follows: given a training data set, sampled from a probability density function $p(x)$, find an approximation (estimation) of $p(x)$ [1]. Perhaps the most famous method for density estimation is the Parzen windows or kernel density estimator, which builds an approximation of $p(x)$ as a mixture of density components, called kernel functions, using all the training data [1]. There exists methods for computing density estimations as the Parzen windows do but without using the whole training data set, the reduced set density estimator [2] follows such an approach by reducing the problem to that of selecting a reduced set of training data as the support components of the estimation, along with the proper weights of those components in the mixture [2].

Stibor and Timmis [3] proposed a quality measure for the aiNet algorithm based on the Parzen window estimator. Alonso et al. [4] put artificial immune network (AIN) field in the context of density estimation and designed also a quality measure based on likelihood functions by relating antibodies to mixture components in a Parzen window estimator. This paper builds on the idea of considering AIN models as density estimators and proposes a new AIN model. Again, each antibody is considered as the mean of a Gaussian kernel function determined by a linear combination of stimulating antigens, which leads to a redefinition of the network dynamics and metadynamics. The model is designed so that it be a reduced set density estimator, where mixture weights are related to antibody stimulation levels.

The paper is organized as follows: Sect. 2 reviews the problem of density estimation and discusses the immune network metaphor as a density estimation method; Sect. 3

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 254–265, 2008.

presents an AIN model for density estimation based on the Parzen window estimator; Sect. 4 shows results from some exploratory experiments that were carried out in order to validate the proposed model, and finally, Sect. 5 concludes the paper and presents some ideas for future work.

2 Density Estimation and AIN

2.1 The Density Estimation Problem

The problem of density estimation can be defined as follows: Given an independent and identically distributed (iid) sample $X = \{x^t\}_{t=1}^N$ drawn from an unknown probability density function $p(x)$, find an approximation (estimation), $\hat{p}(x)$, of $p(x)$ [1]. For this purpose, a likelihood criterion is used, i.e., we search $\hat{p}(x)$ that makes sampling X from it as likely as possible [1]. Provided data in X are iid, the likelihood function of X given the estimation $\hat{p}(x)$ is

$$l(\hat{p}(\cdot)) \equiv \hat{p}(X) = \prod_{t=1}^{N} \hat{p}\left(x^t\right) \quad . \tag{1}$$

As we are interested in finding the $\hat{p}(\cdot)$ that maximizes $l(\cdot)$ given the sample X, this function is usually denoted as $l(\hat{p}(\cdot)|X)$ [1]. In practical applications the log of the likelihood function is maximized in order to obtain some computational simplifications [1]. The log likelihood function is defined as

$$\mathscr{L}(\hat{p}(\cdot)|X) \equiv \log l(\hat{p}(\cdot)|X) = \sum_{t=1}^{N} \log \hat{p}\left(x^t\right) \quad . \tag{2}$$

The density estimation problem can be approached, at least, from three points of view: the parametric, the semiparametric and the nonparametric [1].

In the parametric approach we assume (or know) that $p(x)$ belongs to a particular probability density family, $p(x|\theta)$, where each member is identified by a set θ of parameter values, hence we look for the θ that identifies the underlying distribution [1]. The assumption of a particular probability density family may correspond to assuming that the sample forms a single group, but in many applications the sample may form more than one single group [1]. Then, in the semiparametric approach, we assume that the sample form k groups each of those can be represented by densities belonging to a particular probability density family $p(x|\theta)$ [1]. In such situation, $p(x)$ can be represented as a mixture of such densities [1], which can be written as

$$p(x) = \sum_{i=1}^{k} p(x|\theta_i) P(\theta_i) \quad . \tag{3}$$

where $p(x|\theta_i)$ is the model of the i-th group, also called a mixture component, and $P(\theta_i)$ is the proportion in the mixture of the i-th group. In this setting, we have to estimate both the parameters of each mixture component and the proportion for each one. Notice that k is a hyperparameter that should be specified beforehand [1].

In the nonparametric approach, the underlying assumption is just that similar inputs have similar outputs [1]. Then the estimator looks for similar training data, using a

particular similarity function, and then interpolates from them to give the density value. Differences between nonparametric methods come from the similarity measure or the interpolation strategy [1].

The most popular nonparametric method is the so called Parzen window method or kernel estimator [1]. This method looks for getting a smooth estimate by using a smooth weight function, called kernel function [1]. The general form of the kernel estimator [2] is defined as

$$\hat{p}(x) = \sum_{t=1}^{N} \gamma_t K_h\left(x, x^t\right) \quad . \tag{4}$$

where the kernel function $K_h(\cdot)$ determines the shape of the influences, the window width h determines the width, and each γ_t is a coefficient that should be estimated [1]. Notice that following the above definition, all the x^t have an effect on the estimation of x. If $K_h(\cdot)$ happens to be a density function, so will $\hat{p}(\cdot)$ be [1], subject to the constraint $\sum_{t=1}^{N} \gamma_t = 1$ and $\gamma_t \geq 0, t = 1, \ldots, N$. Thus coefficients can be estimated using a maximum likelihood criterion [2] which produces $\gamma_t = \frac{1}{N} \forall x^t \in X$. Perhaps the most used kernel function is the d-dimensional Gaussian kernel

$$K_h\left(x, x^t\right) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \exp\left[-\frac{\|x - x^t\|^2}{2h^2}\right] \quad . \tag{5}$$

Nonparametric methods have the disadvantage of an increased need for memory and computation since the model is made up by the whole training data set, which could be too expensive for large data sets [2]. In order to overcome this problem, two approaches have been adopted [2]: one approach provides an approximation of $K_h\left(x, x^t\right)$ composed of two functions, one depending only on the point x and the other one depending only on the sample points x^t [2]. The other approach provides an estimate of $p(x)$ by using a reduced set of (representative) sample points. See [2] for a review of both approaches. In [2] the reduced set density estimator (RSDE) is proposed as a method following the second approach. The derivation of the RSDE follows from a quadratic programming problem stated in terms of the general form of the kernel density estimator, whose solution yields a sparse representation in the weighting coefficients and the estimation can be written as

$$\hat{p}(x; h, \gamma) = \sum_{m=1}^{M} \gamma_m K_h(x, x_m) \quad . \tag{6}$$

where $M < N$. The set $\{x_1, \ldots, x_M\}$ is the reduced set of the RSDE [2].

RSDE reduces the computational cost of the original kernel density estimation with similar density accuracy [2]. In [5] a fast version of the RSDE is presented in order to improve the time and space complexities of RSDE that can be challenging for practical applications [5].

2.2 AIN as a Reduced Set Density Estimator

Artificial immune network models have been applied mainly to data clustering [6]-[12], function optimization [13] and autonomous navigation [14]-[17] problems. Regarding

the AIN models for clustering, we can consider the immune network metaphor as a density estimation method, since the basic idea of those models is to locate antibodies in dense zones of antigens, where antibody location searching is based on the affinity maturation process [18]. In recent works, the idea of considering density estimation as an application domain for AIN has been implicitly and explicitly mentioned. Nasraoui et al. [7], [8] designed stimulation functions and B-cell influence zones based on Gaussian kernel functions; Stibor and Timmis [3] proposed a measure for the compression quality of aiNet based on the Parzen window method, and Alonso et al. [4] proposed a general quality measure for AIN performance using the log-likelihood measure. Alonso et al. [4] explicitly stated that the problem solved by AIN is semi-parametric density estimation.

In several AIN models, each antibody b (or B-cell) is represented as a point in a $d-$dimensional space along with a recognition scope r, which defines a hypersphere with center b and radius r. Each antibody may be seen as a sample point in an RSDE model. In this case the density estimation function is given by

$$\hat{p}(x) = \sum_{b_m \in B} \gamma_m K_h(b_m, x) \quad . \tag{7}$$

where B is the set of antibodies and h is related to the antibody radius.

With such interpretation of antibodies, the question of how to find the weights of each antibody still remains. Being affinity a measure of the similarity between antibodies and antigens (in the shape-space), we can consider the antibody's shape as defined by the set of antigens it has recognized, where the higher the affinity the greater influence on the antibody shape an antigen does. Notice that in most of AIN models, when an antibody (or B-cell) recognizes an antigen, the antibody gets stimulated, where the stimulation factor depends on the affinity with the antigen. As mentioned before, antibody's shape is defined by the set of antigens it has recognized, then, it is natural to think that weights for each antibody can be calculated from its stimulation level, specifically, if s_i denotes the stimulation level of antibody b_i then, the weight of each antibody can be computed as

$$\gamma_i = \frac{s_i}{\sum_j s_j} \quad . \tag{8}$$

3 An Artificial Immune Network Density Estimator

Previously proposed AIN models did not have the density estimation problem as a goal. This means that their performance in solving this problem could be improved if they were designed having this goal in mind.

In this section, we present the INDIE (Immune Network DensIty Estimation) algorithm. We follow the idea of the reduced version of the Parzen window estimator [2] INDIE selects a reduced set of data (antigens) for computing the estimation. Following the immune metaphor, antigens determine the shapes of the antibodies, since the immune system tries to generate antibodies specific to the invading antigens [19].

According to De Castro and Timmis [20], an artificial immune network can be described in terms of its structure, dynamics and metadynamics. Structure includes representation of antigens and antibodies and the kind of interactions considered between

them. Dynamics refers to the learning process mainly describing the affinity maturation process. Metadynamics involves mechanisms for natural birth and death of immune elements [20]. Taking into account this requirement and the relationship established in Sect. 2 between AIN and the density estimation problem, the INDIE algorithm is presented.

3.1 INDIE's Structure

The network will be composed of antibodies. The shape of each antibody will be determined by a reduced set of antigens, i.e., an antibody b_i will be represented as

$$b_i = \sum_{j=1}^{k_i} \gamma_{ij} a_j^{(i)} \quad . \tag{9}$$

where k_i is the number of antigens that determine the shape of the antibody b_i. There is a global saturation limit k, thus $k_i \leq k$.

An antibody gets stimulated by an antigen whenever recognition takes place. Stimulation and suppression interactions between antibodies are implicitly modeled and will be discussed below. Additionally, for density estimation purposes we consider that the model is built upon a mixture of Gaussian kernels $K_\sigma(b_i, \cdot)$.

3.2 Network Dynamics

The INDIE algorithm can perform in an on-line fashion. Then, the network dynamics will be defined as the response of the network to the invasion of a single antigen. We distinguish between primary response and secondary response based on thresholds defined upon the kernel parameter σ: an antigen a is recognized by the antibody b if $\|a-b\| < 3\sigma$; a falls in the low affinity zone of b if $2\sigma \leq \|a-b\| < 3\sigma$; and a falls in the high affinity zone of b if $\|a-b\| < 2\sigma$. Let B be the repertoire of antibodies and let a be a new antigen entering the system:

Primary response: There will be two scenarios of primary response:

1. Antigen a is not recognized by any antibody in B, i.e., $\|a-b\| \geq 3\sigma, \forall b \in B$
   In this case, the system respond by creating a new antibody b_{new} at the a location, i.e.,
   $$b_{\text{new}} = 1 \cdot a \quad . \tag{10}$$
2. Antigen a is recognized by some antibodies but recognition happens with low affinity, i.e., $\exists b \in B$ s.t. $2\sigma \leq \|a-b\| < 3\sigma$
   In this case we need to consider two events. Let $B(a)$ be the set of antibodies recognizing a, and let $B^*(a)$ be the set of antibodies recognizing a whose shapes are already determined by k antigens:
   (a) The first event to consider is when at least one b in $B(a)$ is defined by less than k antigens, i.e., $B^*(a) \subset B(a)$
   $\neq$
   In this case, the antibody $b^* \in B(a) - B^*(a)$ with the highest affinity is selected to perform affinity maturation.

(b) The second event to consider is when all the antibodies in $B(a)$ are defined by exactly k antigens, i.e., $B^*(a) = B(a)$
In this case a new antibody b_{new} is created at the a location, i.e., $b_{\text{new}} = 1 \cdot a$ and is linked to the antibody $b^* \in B(a)$ (this could be interpreted as an affinity maturation result)

Secondary response: Secondary response happens when $\exists b \in B$ s.t. $0 \leq \|a-b\| < 2\sigma$. In this case, the system does nothing, which means that a is already covered and then the response is fast, specific and more effective.

Affinity maturation: As mentioned above, affinity maturation process is composed of two mechanisms: cloning and somatic hypermutation [18]. But notice that the goal of affinity maturation could be interpreted as affinity optimization. As the affinity is based on distance function, we can implement affinity maturation (cloning and mutation) just by locating the antibody b at the center of mass of the antigens in its linear combination. Another goal of affinity maturation is the transition from primary response scenarios to secondary response scenarios. We can see that the center of mass has the property that any of the antigens in the linear combination will fall in the high affinity zone after reallocating b.

Stimulation: As is usual in the AIN models, a factor that tracks the stimulation history of each antibody is considered. Let $S_t(b)$ be the stimulation level of antibody b after the presentation of the t-th antigen. This stimulation level is updated accordingly to:

$$S_{t+1}(b) = \begin{cases} S_t(b) + \exp\left(\frac{-\|Ag-b\|^2}{\sigma^2}\right) & b \in B(a_{t+1}) \\ S_t(b) - p & b \in B - B(a_{t+1}) \\ 1 & b \text{ is created to neutralize } a_{t+1} \end{cases} . \quad (11)$$

where p is a penalization factor.

Thus, $S_{t+1}(b)$ can be described as

$$S_{t+1}(b) = S_t(b) + S^{\mathbb{A}}(b) + S^{\mathbb{B}}(b) - S_{\text{lost}} . \quad (12)$$

where $S^{\mathbb{A}}(b)$ is the stimulation received by antibody b because of the antigen intrusion, $S^{\mathbb{B}}(b)$ is the stimulation received by antibody b because of its linking to other antibodies, and S_{lost} is a constant stimulation lost factor. Notice that in this approach, antibody suppression is implicitly defined.

3.3 Metadynamics

In the metadynamics step, all the antibodies b such that $S_t(b) \leq 0$ are removed from the network. No new antibodies cells are randomly generated.

4 Exploratory Experiments

In this section we explore the learning capabilities of the INDIE algorithm. For such purpose we use the log-likelihood function as a performance measure accordingly to the density estimation problem. We run the algorithm on the following data sets:

1. Two Gaussian clusters data set: a bi-dimensional data set composed of 500 instances, The clusters were generated using the SSJ library[1] with $\mu_1 = (8.0, 2.0)$, $\mu_2 = (2.0, 8.0)$ and $\Sigma_1 = \Sigma_2 = I_2$.
2. Two spirals data set: a bi-dimensional data set composed of 190 instances. The points were generated using the spir.m script of the aiNet package[2].
3. Iris data set: the well known iris 4-dimensional data set from the UCI Machine Learning Repository[3], composed of 150 instances.
4. Two noisy Gaussian clusters data set: a bi-dimensional data set composed of 1018 instances.

The first experiment was carried out using data sets 1-3 in order to compare the likelihood of the sample given the INDIE's estimation to the likelihood of the sample given the real and/or the uniform distribution. In all the cases, we have to consider the saturation limit k, the kernel parameter σ, and the penalization factor p as the parameters of INDIE. Table 1 shows the parameter values used for each data set.

Table 1. INDIE parameter values

Data set	k	σ	p
Two Gaussian clusters	20	0.5	0.001
Two spirals	3	0.1	0.001
Iris	10	0.5	0.001

Figure 1 shows the evolution of the log-likelihood while antigens are presented to the network. The figure also shows the log-likelihood of the sample given a uniform distribution and the log-likelihood of the sample given the real distribution when it is known, which is the case of the two Gaussian clusters data set.

Notice that in the three cases, the log-likelihood of the sample given the INDIE's estimation increase as more antigens are presented to the network, which is an expected behavior of a density estimator: the greater the sample size, the more reliable the estimation. Notice also that in the two Gaussian clusters and Iris data sets, INDIE achieves a better estimation than the uniform distribution, while in the case of the two spiral data set INDIE's estimation is below the uniform.

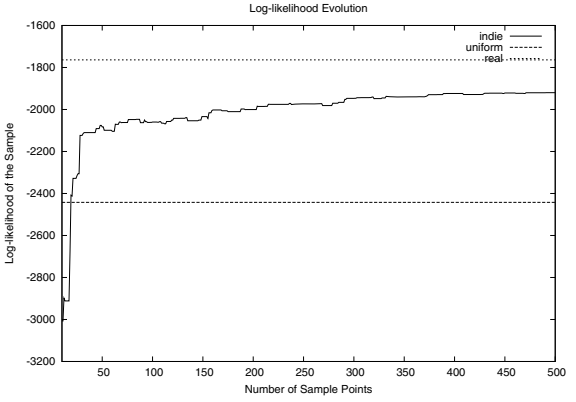
Figure 2 shows the antibody location for the two Gaussian clusters and the two spirals data set. This has the purpose of a visual validation.

Notice that in both cases, antibodies are located at dense zones of antigens. This suggests also, that INDIE's estimation can be used for clustering purposes using a likelihood labeling strategy. This direct strategy may not work well on the two spirals data set, but again it is due to the structure of the space. If it is possible to transform the input space to another feature space (for example using the strategy of kernel methods [21]) these results may be improved.

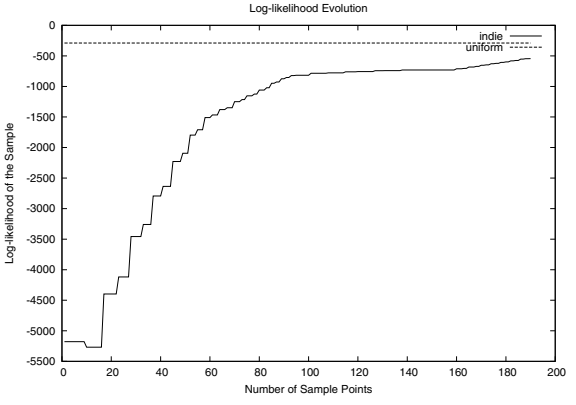
[1] Available at http://www.iro.umontreal.ca/∼simardr/ssj/indexe.html `http://www.iro.umontreal.ca/~simardr/ssj/indexe.html`

[2] Available at http://www.dca.fee.unicamp.br/ lnunes/manual.html `http://www.dca.fee.unicamp.br/~lnunes/manual.html`

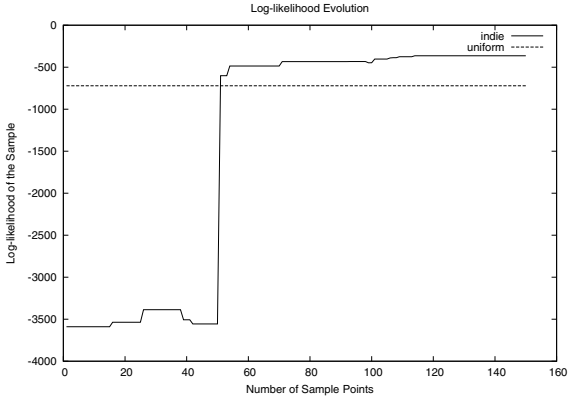
[3] http://archive.ics.uci.edu/ml/ `http://archive.ics.uci.edu/ml/`



(a) two gaussian clusters

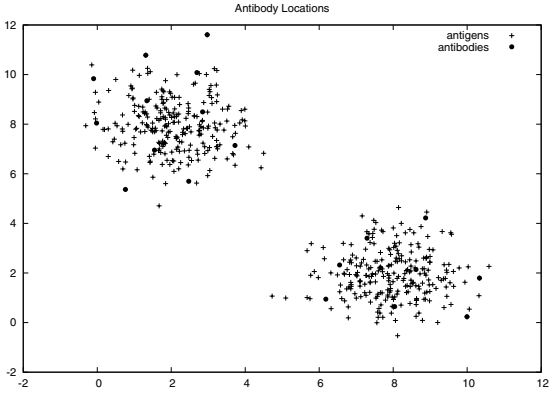


(b) two spirals

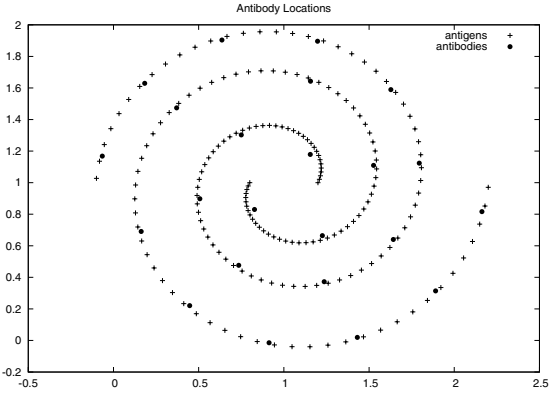


(c) iris

Fig. 1. Log-likelihood evolution. (a) two Gaussian clusters; (b) two spirals; (c) iris.



(a) two gaussian clusters



(b) two spirals

Fig. 2. Antibody locations. (a) two Gaussian clusters; (b) two spirals.

Table 2. Amount of antibodies and support antigens: MNAPA means Maximum Number of Antigens per Antibody, while TNSA means Total Number of Support Antigens

Data set	Two Gaussian clusters	Two spirals	Iris
Antibodies	19	23	5
MNAPA	15	3	5
TNSA	90	69	14

In order to give some insights of the INDIE capabilities for building a reduced set density estimation, Table 2 shows the number of antibodies at the end of the process, along with the maximum and total number of antigens determining the shape of such antibodies.

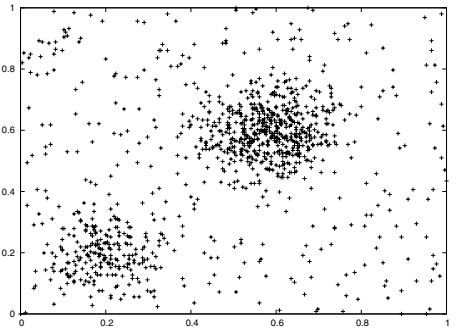


Fig. 3. Two noisy Gaussian clusters data set

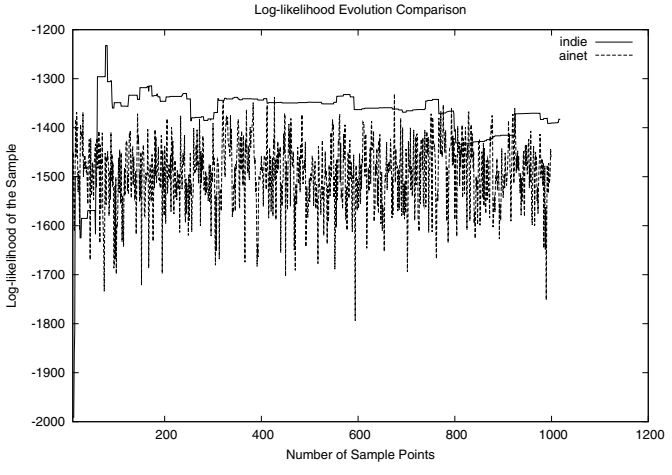


Fig. 4. Log-likelihood evolution: aiNet vs INDIE

The second experiment was carried out in order to compare the INDIE algorithm to the aiNet [6] algorithm in a density estimation task. For this purpose the data set 4 was used (see Fig. 3). The parameters of the aiNet algorithm in this setting were $ts = 0.1$, $n = 4$, $N = 10$, $qi = 0.1$, $tp = 1$ and $gen = 1$, this last parameter was set to 1 to simulate an on-line environment for the aiNet algorithm. The parameters of the INDIE algorithm were $k = 40$, $\sigma = 0.1$ and $p = 0$. Figure 4 shows the evolution of the log-likelihood given by both algorithms. Notice that the estimation given by INDIE is more stable and better than the estimation given by the aiNet algorithm, This is expected since INDIE was designed specially for the density estimation problem, while aiNet was not. From the picture can be concluded also, that INDIE is less sensible to noisy data than aiNet. It is worthy to remark that for this estimation aiNet uses 40 antibodies, while INDIE uses only 9.

5 Conclusions and Future Work

A new artificial immune network model was proposed. The model is designed having in mind the problem of nonparametric density estimation. A new representation for antibodies is presented, which is motivated by considering that antibody shapes are determined by the stimulating antigens following the affinity maturation process. We also exploit the on-line learning capability of immune systems by designing the model so that each antigen is presented only once.

Some exploratory experiments were carried out on synthetic and real data sets and the log-likelihood function was used as a performance measure comparing the INDIE estimation with the uniform distribution and/or the real distribution. Experimental results suggest that the proposed model can be set up so that the reliability of the estimation increases with the number of samples approaching the real distribution.

An important feature of the proposed model is that it is very simple relying only on distance functions, which suggests that it can be applied in a similarity-based setting where no vector representation is assumed but a (dis)similarity function is available. Taking into account those features we proposed the following as a future work:

- To apply the model in a pure similarity-based learning task.
- To present the model as a kernel method [21] so that it can be applied in feature spaces induced by kernel functions.
- To characterize the proposed model taking into account stability and convergence on sample orderings.
- To develop a method for the parameter tuning.

References

1. Alpaydin, E.: Introduction to Machine Learning. The MIT Press, Cambridge (2004)
2. Girolami, M., He, C.: Probability density estimation from optimally condensed data samples. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 1253–1264 (2003)
3. Stibor, T., Timmis, J.: An investigation into the compression quality of aiNET. In: Foundations of Computational Intelligence. IEEE Symposium on Computational Intelligence, HI, USA, pp. 495–502 (2007)
4. Alonso, O., González, F.A., Niño, F., Galeano-Huertas, J.C.: A solution concept for artificial immune networks: A coevolutionary perspective. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 35–46. Springer, Heidelberg (2007)
5. Deng, Z., Chung, F.L., Wang, S.: FRSDE: Fast reduced set density estimator using minimal enclosing ball approximation. Pattern Recognition 41, 1363–1372 (2008)
6. de Castro, L.N., Zuben, F.J.V.: aiNet: An Artificial Immune Network for Data Analysis. In: Abbas, H.A., Sarker, R.A., Newton, C.S. (eds.) Data Mining: A Heuristic Approach, pp. 231–259. Idea Group Publishing, USA (2001)
7. Nasraoui, O., Cardona, C., Rojas, C., González, F.: TECNO-STREAMS: Tracking evolving clusters in noisy data streams with a scalable immune system learning model. In: Third IEEE International Conference on Data Mining, Melbourne, FL. IEEE, Los Alamitos (2003)
8. Nasraoui, O., González, F., Dasgupta, D.: The Fuzzy Artificial Immune System: Motivations, Basic Concepts and Application to Clustering and Web Profiling. In: IEEE International Conference on Fuzzy Systems, Hawaii, HI, pp. 711–716. IEEE, Los Alamitos (2002)

9. Neal, M.: An Artificial Immune System for Continuous Analysis of Time-Varying Data. In: Timmis, J., Bentley, P.J. (eds.) Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS), vol. 1, pp. 76–85. University of Kent at Canterbury, University of Kent at Canterbury Printing Unit (2002)
10. Neal, M.: Meta-Stable Memory in an Artificial Immune Network. In: Timmis, J., Bentley, P., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 168–180. Springer, Heidelberg (2003)
11. Timmis, J., Neal, M.: A Resource Limited Artificial Immune System for Data Analysis. Knowledge-Based Systems 14, 121–130 (2001)
12. Timmis, J., Neal, M., Hunt, J.: An Artificial Immune System for Data Analysis. BioSystems 55, 143–150 (2000)
13. de Castro, L.N., Timmis, J.: An Artificial Immune Network for Multimodal Optimisation. In: Congress on Evolutionary Computation. Part of the 2002 IEEE World Congress on Computational Intelligence, Honolulu, Hawaii, USA, pp. 699–704. IEEE, Los Alamitos (2002)
14. Ishiguro, A., Ichikawa, S., Uchikawa, Y.: A Gait Acquisition of Six-Legged Robot Using Immune Networks. In: Proceedings of International Conference on Intelligent Robotics and Systems (IROS 1994), Munich, Germany, vol. 2, pp. 1034–1041 (1994)
15. Luh, G.C., Liu, W.W.: Reactive Immune Network Based Mobile Robot Navigation. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) Proceeding of the Third Conference ICARIS, pp. 119–132. Springer, Heidelberg (2004)
16. Michelan, R., Zuben, F.J.V.: Decentralized Control System for Autonomous Navigation Based on an Evolved Artificial Immune Network. In: Proceedings of the IEEE Congress on Evolutionary Computation, Honolulu, HI, vol. 2, pp. 1021–1026. IEEE, Los Alamitos (2002)
17. Vargas, P.A., de Castro, L.N., Michelan, R., Zuben, F.J.V.: An Immune Learning Classifier System for Autonomous Navigation. In: Timmis, J., Bentley, P., Hart, E. (eds.) Proceedings of the Second International Conference ICARIS, Edinburg, UK, pp. 69–80. Springer, Heidelberg (2003)
18. Galeano, J.C., Veloza-Suan, A., González, F.A.: A comparative analysis of artificial immune network models. In: Beyer, H. (ed.) Proceedings of the Genetic and Evolutionary Computation Conference, pp. 361–368. ACM Press, New York (2005)
19. Perelson, A.S., Weisbuch, G.: Immunology for Physicists. Reviews of Modern Physics 69, 1219–1267 (1997)
20. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)
21. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)

Learning Probability Densities of Optimization Problems with Constraints and Uncertainty

Jonás Velasco, Aydeé López Muñiz, and Arturo Berrones

Posgrado en Ingeniería de Sistemas
Centro de Innovación, Investigación y Desarrollo en Ingeniería y Tecnología
Universidad Autónoma de Nuevo León
AP 126, Cd. Universitaria, San Nicolás de los Garza, NL 66450, México
arturo@yalma.fime.uanl.mx

Abstract. A recently introduced method for the learning of the asymptotic marginal densities of stochastic search processes in optimization is generalized for its application in problems under constraints and uncertainty. The use of the proposed approach as a mechanism for diversification in optimization algorithms is illustrated on several benchmark examples.

1 Introduction

Constraints and uncertainty are two important aspects that arise in real–world applications, which may impose special difficulties to optimization algorithms. For instance, no universal framework exists for constraint handling in population based algorithms [1]. A common approach to optimization problems under uncertainty, on the other hand, is based on variations of Monte–Carlo methods, which imply the approximation of high dimensional integrals at a usually large computation cost [9]. In this contribution a recently introduced approach for density estimation in stochastic search is extended to cases which involve constraints and uncertainty. It's argued that the proposed method could be used like a suitable diversification strategy in connection with local search procedures.

The present work is based on the Stationary Fokker–Planck Learning (SFPL) technique for the estimation of stationary marginal densities of stochastic search processes [2,3,4]. In its original formulation, the minimization of a cost function of the form $V(x_1, x_2, ..., x_n, ..., x_N)$ with a search space defined over a box $L_{1,n} \leq x_n \leq L_{2,n}$ is considered. In this contribution the SFPL technique is generalized to consider arbitrary constraints. Moreover, the SFPL algorithm is for the first time tested on a problem with uncertain parameters. In order to proceed, a brief survey of SFPL is first presented. A stochastic search process for the minimization of V is modeled by

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

where $\varepsilon(t)$ is an additive noise with zero mean. Equation (1), captures the basic properties of a general stochastic search strategy. Under an uncorrelated Gaussian noise with constant strength, Eq. (1) represents a search by diffusion, while

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 266–272, 2008.

a noise strength that is slowly varying in time gives a simulated annealing. Notice that when choosing an external noise of infinite amplitude, the dynamical influence of the cost function over the exploration process is lost, leading to a blind search. The model given by Eq.(1) can be also interpreted as an overdamped nonlinear dynamical system composed by N interacting particles. The temporal evolution of the probability density of such a system in the presence of an additive Gaussian white noise, is described by a linear differential equation, the Fokker – Planck equation [5,6]. The SFPL method is based on the representation of the cumulative conditionals of the stationary solution of this equation by means of a complete set,

$$\hat{y}(x_n|\{x_{j\neq n}\}) = \sum_{l=1}^{L} a_{n,l}\varphi_l(x_n). \tag{2}$$

The specification of the free parameters a's come from the imposed boundaries and from the evaluation of the stationary Fokker – Planck equation for the cumulative conditionals,

$$\frac{d^2y}{dx_n^2} + \frac{1}{D}\frac{\partial V}{\partial x_n}\frac{dy}{dx_n} = 0, \tag{3}$$

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

The equation (3) is evaluated in the required number of interior points in order to have a linear system of algebraic equations for the a's. Starting from an arbitray point in the domain, from the resulting marginals a new point is drawn, which now has information on the objective function V. It's shown in [3] that by the iteration of this procedure, the marginal cumulative distribution for each variable can be estimated by

$$\langle\hat{y}\rangle = \sum_{l=1}^{L} \langle a_{n,l}\rangle\, \varphi_l(x_n) \rightarrow y(x_n), \tag{4}$$

where the brackets represent the average over the iterations.

2 Constraint Handling

Under the SFPL framework, constraints are introduced using the notion of a "region with low probability". This means that, during the iterations of the SFPL algorithm, the marginal densities should learn to associate low probability values to regions outside the valid domain. The notion can be implemented in different ways. For instance, suitable nonlinear terms may be added to the cost function. The usefulness of this approach has been already discussed in [3]. On that work, constraints enter through the construction of "force barriers" in terms of smooth nonlinear functions. This leads to the introduction of a number

of additional free parameters that control the shape of the nonlinearities. A different approach is proposed here. This consists on the definition of a new objetive function associated with problem's constraints. It's shown that by this method, SFPL is capable to learn a probability density that closely represents the valid region. Consider the optimization problem

$$\begin{aligned} min \quad & f(x) \\ s.t. \quad & h_1(x) = 0, ..., h_I = 0 \\ g_1(x) < 0, ..., g_R(x) < 0, \quad & L_{1,n} \leq x_n \leq L_{2,n}. \end{aligned} \tag{5}$$

In order to proceed, the following transformation to a biobjective model suggested in [7] is used

$$\begin{aligned} min \quad & f(x) \\ s.t. \quad & L_{1,n} \leq x_n \leq L_{2,n} \end{aligned} \tag{6}$$

$$\begin{aligned} min \quad & w(x) = \sum_{i=1}^{I} |h_i| + \sum_{r=1}^{R} S(g_r) g_r \\ s.t. \quad & L_{1,n} \leq x_n \leq L_{2,n} \end{aligned} \tag{7}$$

where S is a step function such that $S(g_r < 0) = 0$ and $S(g_r > 0) = 1$. By construction the minimum of w is $w = 0$ and is located where the constraints are satisfied.

The SFPL is capable to learn probability densities that describe the valid region of a constrained problem through the direct application of the SFPL algorithm to the objective function w. This is illustrated by the following benchmark problems which have a very small valid region, as discussed in [8].

Test Problem 1:

$$\begin{aligned} min \quad & x_1^2 + (x_2 - 1)^2 \\ s.t. \quad & x_2 - x_1^2 = 0, \\ -1 \leq x_1 \leq 1, \quad & -1 \leq x_2 \leq 1, \end{aligned}$$

solution: $(\pm 1/\sqrt{2}, 1/2)$.

Test Problem 2:

$$\begin{aligned} min \quad & e^{x_1 x_2 x_3 x_4 x_5} \\ s.t. \quad & x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0, \\ & x_2 x_3 - 5 x_4 x_5 = 0, \\ & x_1^3 + x_2^3 + 1 = 0, \\ & -2.3 \leq x_i \leq 2.3 \quad (i = 1, 2), \\ & -3.2 \leq x_i \leq 3.2 \quad (i = 3, 4, 5), \end{aligned}$$

solution: $(-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$.

Table 1. Results for the Test Problem 1 and Test Problem 2 with a tolerance of $\epsilon = 0.001$ and $\epsilon = 0.01$ respectively. A ratio is estimated between feasible solutions and the total number of randomly generated solutions taken from SFPL and uniform densities.

	Iterations	**SFPL density**	**Uniform density**
Test Problem 1	100	0.29%	0.02%
	500	0.32%	0.06%
Test Problem 2	100	0.61%	0.00%
	500	0.556%	0.00%

A comparision between uniform densities and the densities generated by SFPL is presented in Table 1. The experiment consists in drawing points from the densities in order to evaluate the ratio of points for which $w < \epsilon$ with respect to the total number of drawn points. The SFPL algorithm uses the parameter values of $L = 200$ and $D = 0.03$. The results indicate that SFPL gives a substantial improvement with respect to the use of uniform densities in both test problems. The potential use of SFPL like a diversification mechanism in constrained problems is shown in Fig. 1. The figure presents the the product $p(x_n|f)p(x_n|w)$ of the densities learned by SFPL for objectives f and w in Test Problem 1. Is clear that this joint density will have large values in regions that satisfy the constraints and improve the cost. For the presented example, the parameters of the SFPL are $L = 100$, $D = 0.5$ for the estimation of $p(x_n|f)$ and $L = 200$, $D = 0.03$ for the estimation of $p(x_n|w)$. In order to generate the graph presented in Fig. 1 The algorithm ran over 1 and 10 iterations respectively (only one iteration is necessary for the cost function by the fact that this is a separable problem, as discussed in [3]). Using the formulas given in [3], it turns out a total computation effort of 396 cost function evaluations and 7960 constraint function evaluations.

The SFPL algorithm can be linked to an adaptation mechanism. For instance, the development of a genetic algorithm in which the population diversity is taken from the regions of maximum probability is currently a work in progress.

3 SFPL Approach to Uncertainty

Because of its stochastic formulation, the SFPL algorithm appears to naturally adapt to problems which involve uncertainty in the cost function. This expectation is for the first time tested on the well known newsvendor problem, for which a closed general expression for the solution is known. This solution is used to test the quality of the estimation given by SFPL. The newsvendor problem is formulated in the following way [9],

$$max \quad E[q \quad min(x, d)] - cx \tag{8}$$

This model is used to determine optimal inventory levels. Suposse that x is the quantity of raw materials at the warehouse of a production center. The parameter q gives the market price of each unit of raw material transformed into

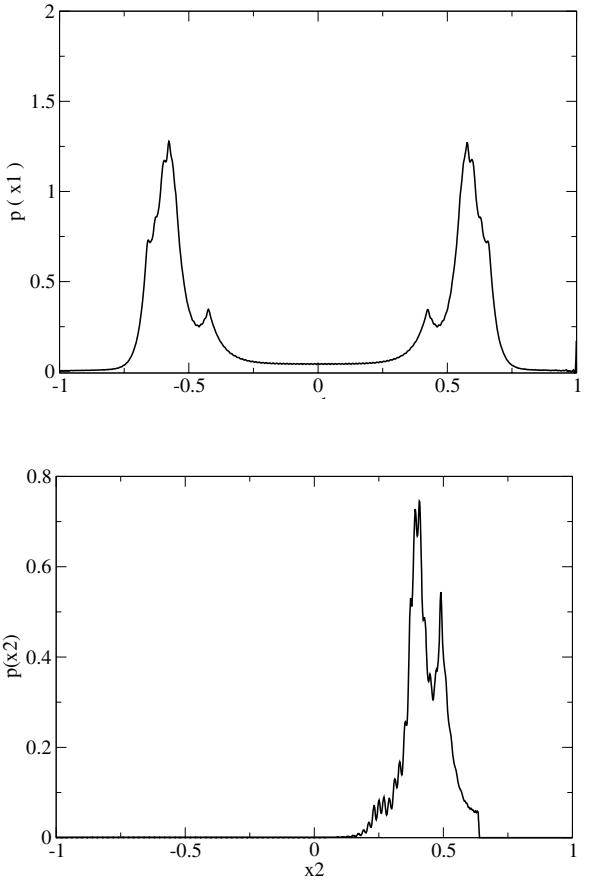


Fig. 1. The joint density $p(x_n|f)p(x_n|w)$ of test problem 1. The region of high probability of the learned density corresponds to the known global optimum.

a given product. The parameter c represents the production cost per unit of raw material. The parameter d is a random demand, distributed according to some cumulative distribution $F(d)$. The problem is to find x, in order to maximize the expected profit. The analytical solution of this problem is given by [9]

$$x = F^{-1}\left(\frac{q-c}{q}\right), \tag{9}$$

where F^{-1} denotes the inverse of the cumulative distribution of d. In order to apply the SFPL method to this problem, consider the equivalent formulation

$$min \quad cx - E[q \quad min(x, d)] \tag{10}$$

An experiment with uniformily distributed demand is reported in Fig. 2 At each gradient evaluation a fixed demand d is passed to the cost function, which is evaluated by

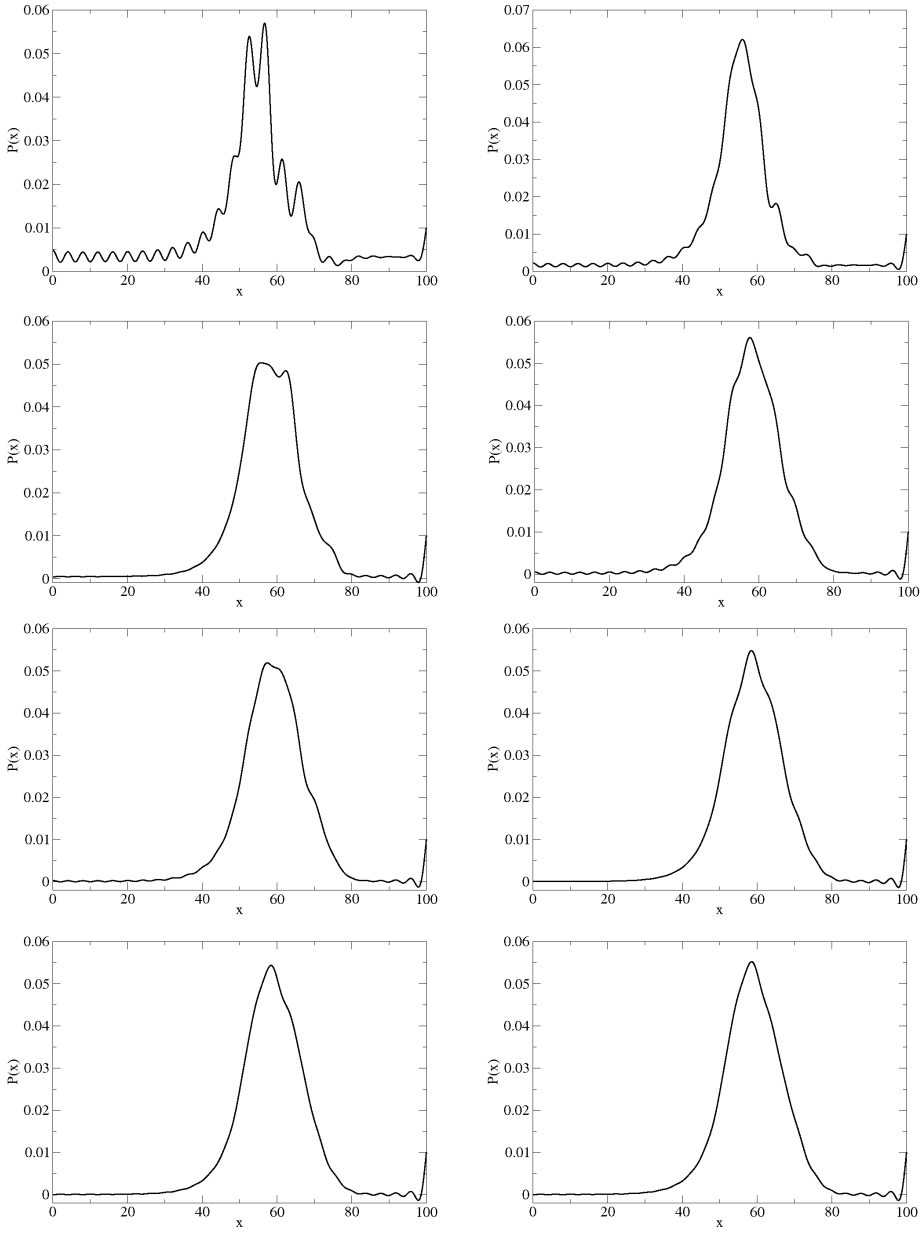


Fig. 2. Probability densities of the Newsvendor problem. Iterations 2, 5, 20, 50, 100, 200, 500, 800 of the SFPL algorithm are plotted, with SFPL parameters L=50 and D=10.

$$\begin{aligned} if(x < d) \quad & value = cx - qx \\ else \quad & value = cx - qd \\ return \quad & value \end{aligned}$$

The Fig. 2 indicates convergence to the correct distribution in a problem with $p = 7$, $c = 5$ and demand uniformly distributed in the interval $[50, 80]$. The region of high probability is entirely consistent with the known optimal solution, given by $x = 58.55$.

4 Conclusions

The numerical experiments indicate that the estimation of the stationary density of random search processes in constrained and stochastic optimization problems can be performed by the SFPL algorithm. The density gives a natural mechanism for diversification. It provides quantitative probability measures for the likelihood of finding good candidate solutions in a given region of the search space. A next natural step in this research is to extend the experimentation to larger scale problems, integrating the procedure with an adaptation mechanism.

References

1. Michalewicz, Z.: A survey of constraint handling techniques in evolutionary computation methods. In: Proc. of the 4th Annual Conf. on Evolutionary Programming, pp. 135–155. MIT Press, Cambridge (1995)
2. Berrones, A.: Generating Random Deviates Consistent with the Long Term Behavior of Stochastic Search Processes in Global Optimization. In: Sandoval, F., Gonzalez Prieto, A., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 1–8. Springer, Heidelberg (2007)
3. Berrones, A.: Stationary probability density of stochastic search processes in global optimization. J. Stat. Mech., P01013 (2008)
4. Peña, D., Sánchez, R., Berrones, A.: Stationary Fokker–Planck Learning for the Optimization of Parameters in Nonlinear Models. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 94–104. Springer, Heidelberg (2007)
5. Risken, H.: The Fokker–Planck Equation. Springer, Berlin (1984)
6. Van Kampen, N.G.: Stochastic Processes in Physics and Chemistry. North-Holland, Amsterdam (1992)
7. Osyczka, A., Krenich, S.: Evolutionary Algorithms for Global Optimization. In: Pintér, J. (ed.) Global Optimization, Scientific and Engineering Case Studies Series: Nonconvex Optimization and Its Applications, vol. 85 (2006)
8. Pulido, G.T., Coello, C.A.C.: A constraint-handling mechanism for particle swarm optimization. In: Proceedings of the 2004 congress on evolutionary computation, vol. 2, pp. 1396–1403 (2004)
9. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer, Berlin (1997)

Pattern Classification Based on Conformal Geometric Algebra and Optimization Techniques

Benjamín Cruz, Ricardo Barrón, and Humberto Sossa

Centro de Investigación en Computación-IPN, Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal
México City, 07738. México
benjamincruz@sagitario.cic.ipn.mx, {rbarron,hsossa}@cic.ipn.mx

Abstract. Conformal Geometric Algebra (CGA) is a high level language commonly used in mathematical, physics and engineering problems. At a top level, CGA is a free coordinate tool for designing and modeling geometric problems; at a low level CGA provides a new coordinate framework for numeric processing in problem solving. In this paper we show how to use quadratic programming and CGA for, given two sets p and q of points in $\mathbb{R}^n$, construct an optimal separation sphere S such that, all points of p are contained inside of it, and all points of q are outside. To classify an unknown pattern x, an inner product must be applied between x and S. Some numerical and real examples to test the proposal are given.

Keywords: Conformal Geometric Algebra, Pattern Classification, Optimization.

1 Introduction

In the XIX century many scientists worked in the development of many algebraic systems. Among these, William K. Clifford (1845-1879) presented the now so-called Geometric Algebras (GA) or Clifford Algebras [1]. These were completely described in his paper "Applications of Grassmann's Extensive Algebra" [2].

A geometric algebra is a free coordinate geometric schema [3]. In GA, the geometric objects and the operators over these objects are treated in a single algebra [4]. A special characteristic of GA is its geometric intuition. In this case, spheres and circles are both algebraic objects with a geometric meaning [1]. Another important characteristic is that, usually the expressions in GA have low symbolic complexity [5].

The Conformal Geometric Algebra (CGA) is a 5-dimensional free coordinate geometric schema; it is a conformal representation for 3D objects. In CGA, points, spheres, and planes are easily represented as vectors. CGA provides a great variety of basic geometric entities to compute with them [5].

2 Basics on Conformal Geometric Algebra

In general, a point in CGA is represented as:

$$P = p + \frac{1}{2}p^2 e_\infty + e_0 \tag{1}$$

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 273–283, 2008.

where p is the Euclidean point, a linear combination of the base vectors (e_1, e_2 and e_3 in the case of 3D space). e_0 and e_∞ representing the 3D origin and the point at infinity respectively, such that $e_0^2 = e_\infty^2 = 0$ and $e_0 \cdot e_\infty = -1$.

Equation (1) expresses a homogeneous relationship between both Euclidean and conformal domains since, given a scalar α and a conformal point P, αP and P represent both the same Euclidean point p. When the coefficient of e_0 is equal to *1*, then P has a canonic representation. In this work, the algebra works in the conformal domain while the geometric semantics lies in the Euclidean domain. In the same way, the sphere takes a canonical form:

$$S = C - \frac{1}{2}\gamma^2 e_\infty = s - \frac{1}{2}\left(c^2 - \gamma^2\right)e_\infty + e_0 \tag{2}$$

where C is the central point in conformal form as defined in (1), with γ is the radius of the sphere and s is the Euclidean center point. A distance measure between one conformal point P and a sphere S can be defined with the help of the inner product [6], as follows:

$$P \cdot S = p \cdot s - \frac{1}{2}\left(s^2 - \gamma^2\right) - \frac{1}{2}p^2 = \frac{1}{2}\left(\gamma^2 - \left(s - p\right)^2\right) \tag{3}$$

where p in simplified form:

$$2\left(P \cdot S\right) = \gamma^2 - \left(s - p\right)^2 \tag{4}$$

Based on this, if $P \cdot S > 0$ then p is inside of the sphere, if $P \cdot S < 0$ then p is outside of the sphere, and if $P \cdot S = 0$ then p is on the sphere.

3 The Classification Problem

Let M be a set of patterns in $\mathbb{R}^n$ and let $m^i \subset M$ a subset of similar patterns each one belonging to i class respectively, where $i = 1, \ldots, q$ and q is the total of classes. Given an unclassified pattern $x \notin M$, the classification problem is to decide to which class it belongs. x must be assigned to the class that contains similar patterns to it. There are many techniques to achieve this. Refer for example [7] and [8].

Two sets of points in *n*-dimensional space can be completely separated by a *n-1*-dimensional hyper-plane, they are said to be *linearly separable*.

Starting from the fact that patterns of a same class are relatively closed in the space, other approximation could be used: the use of support regions can also be used to cluster the space. When two sets of points in n-dimensional space can be completely separated by a $n-1$-dimensional hyper-sphere, they are said to be *spherically separable*.

The goal is to classify a pattern as belonging to a specific class if and only if the pattern is inside of the support region (hyper-sphere) of that class.

The main problem consists on taking all patterns of a specific class and to build a support region that covers all of them efficiently.

In [1], a least squares method is used in order to construct a sphere that fits the points. In this case the drawback is that generally some points might appear located out of the resulting sphere; therefore, the basic classification principle that demands a good classification for the less points used in the training sphere is violated.

In [9] a method to construct an optimal spherical convex hull using quadratic programming and conformal geometric algebra is presented. A convex hull is important because it is a natural way for modeling a class of pattern vectors and, therefore, to classify them according to this class.

4 Proposed Solution

Let $p = \{p_i\}_{i=1}^{n} \cup \{p_j\}_{j=1}^{m}$ a set of points in $\mathbb{R}^n$, the problem is to find an optimal (in quadratics terms) sphere S defined as in (2) with the least square error, such that $\{p_i\}_{i=1}^{n}$ are inside S and $\{p_j\}_{j=1}^{m}$ are outside of it. In other words, to solve:

$$\min_{S} \sum_{i=1}^{n+m} \left(P_i \cdot S\right)^2 \tag{5}$$

subject to $P_i \cdot S \geq 0 \; \forall_{i=1,\ldots n}$ and $P_j \cdot S < 0 \; \forall_{j=1,\ldots m}$, where P_i is the conformal representation of p_i. This problem must be changed in a classical form of quadratic programming:

$$\min_{S} S^t H S + F^t S \tag{6}$$

subject to $AS \leq b$. Let:

$$\begin{aligned}\sum_{i=1}^{n+m}\left(P_i \cdot S\right) &= \sum_{i=1}^{n+m}\left(p_i \cdot s - s_4 - \frac{1}{2}p_i^2\right)^2 = \sum_{i=1}^{n+m}\left(\left[p_i \cdot s - s_4\right] - \frac{1}{2}p_i^2\right)^2 \\ &= \sum_{i=1}^{n+m}\left(\left[p_i \cdot s - s_4\right]^2 - \left[p_i \cdot s - s_4\right]p_i^2 + \frac{1}{4}p_i^4\right)\end{aligned}$$

Thus, the next equation is obtained:

$$\sum_{i=1}^{n+m}\left(P_i \cdot S\right) = \sum_{i=1}^{n+m}\left(p_i \cdot s - s_4\right)^2 + \sum_{i=1}^{n+m}\left(-p_i \cdot s + s_4\right)p_i^2 + \frac{1}{4}\sum_{i=1}^{n+m} p_i^4 \tag{7}$$

In (7) the third term is irrelevant because it does not depend on parameter S and, without lose of generality, (7) can be transformed into:

$$\sum_{i=1}^{n+m}\left(P_i \cdot S\right) = \sum_{i=1}^{n+m}\left(\sum_{k=1}^{4} w_{ik}s_k\right)^2 + \sum_{i=1}^{n+m}\left(\sum_{k=1}^{4} f_{ik}s_k\right) \tag{8}$$

where $w_{ik} = \begin{Bmatrix} p_{i,k} & for\ k = 1,2,3 \\ -1 & for\ k = 4 \end{Bmatrix}$, and $f_{ik} = \begin{Bmatrix} -\left(p_{i,k}\right) p_i^2 & for\ k = 1,2,3 \\ p_i^2 & for\ k = 4 \end{Bmatrix}$.

Let $w_j = \left[w_{ik}\right]$, then (8) can be rewritten:

$$\sum_{i=1}^{n+m}\left(P_i \cdot S\right) = \sum_{i=1}^{n+m}\left(w_i^t S\right)^2 + \sum_{k=1}^{4}\left(\sum_{i=1}^{n+m} f_{ik} s_k\right) \tag{9}$$

Let $F_k = \sum_{i=1}^{n+m} f_{ik}$, then equation (10) is obtained:

$$\sum_{i=1}^{n+m}\left(P_i \cdot S\right) = \sum_{i=1}^{n+m}\left(w_i^t S\right)^2 + \sum_{k=1}^{4}\left(F_k s_k\right) \tag{10}$$

The second term of (10) can be written in matrix notation as:

$$\sum_{k=1}^{4}\left(F_k s_k\right) = F^t S \tag{11}$$

And for the first term of (10) let $j = n + m$:

$$\begin{aligned} \sum_{i=1}^{j}\left(w_i^t S\right)^2 &= \left(w_1^t S\right)^2 + \ldots + \left(w_j^t S\right)^2 \\ &= w_1^t S w_1^t S + \ldots + w_j^t S w_j^t S \\ &= \left(w_1^t S w_1^t + \ldots + w_j^t S w_j^t\right) S \\ &= \left(\left[S^t w_1 \ldots S^t w_j\right] W\right) S \\ &= S^t W^t W S \end{aligned}$$

In summary, equation (10) can be rewritten in matrix notation as:

$$\sum_{i=1}^{n+m}\left(P_i \cdot S\right) = S^t H S + F^t S \tag{12}$$

where $H = W^t W$, which is in the form of (6). The constraint for the points inside the sphere will be:

$$\begin{aligned} P_i \cdot S &< 0 \\ p_i \cdot s - s_4 - \frac{1}{2} p_i^2 &< 0 \\ p_i \cdot s - s_4 &< \frac{1}{2} p_i^2 \\ \sum_{k=1}^{4} w_{ik} s_k &< \frac{1}{2} p_i^2 \end{aligned}$$

where $w_{ik} = \begin{Bmatrix} p_{i,k} & for\ k = 1,2,3 \\ -1 & for\ k = 4 \end{Bmatrix}$, that can be written:

$$W^t S < \frac{1}{2} p_i^2 \tag{13}$$

and the constraint for points outside the sphere will be:

$$\begin{aligned} P_j \cdot S &\geq 0 \\ p_j \cdot s - s_4 - \frac{1}{2} p_j^2 &\geq 0 \\ p_j \cdot s - s_4 &\geq \frac{1}{2} p_j^2 \\ \sum_{k=1}^{4} f_{jk} s_k &\leq -\frac{1}{2} p_j^2 \end{aligned}$$

where $f_{ik} = \begin{Bmatrix} -p_{j,k} & for\ k = 1,2,3 \\ 1 & for\ k = 4 \end{Bmatrix}$, that can be written as:

$$F^t S \leq -\frac{1}{2} p_j^2 \tag{14}$$

Let $A = \left[W, F\right]$ and b is a vector whose *i*-th row is $\frac{1}{2} p_i^2 + \varepsilon$ for $i = 1,\ldots,n$ and the $i + j$ -th row is $-\frac{1}{2} p_j^2$ for $j = 1,\ldots,m$. The ε is a smallest quantity, used to force the inequality (13) will be a $\leq$. Then the constraint already has the form as in (6).

It is clear that by including into the restrictions all the points that stay out of the sphere, the solution S results in a separation surface that allow differentiate between two classes (i.e. inner and outer points). Finally, to find S by solving (6).

Classification of a pattern is performed as: an inner product between an unclassified pattern x and S must be applied:

$$i = S \cdot X \tag{15}$$

where X is the conformal representation of x. When x is inside a class' sphere, equation (15) returns a positive number (or zero), otherwise it returns negative number.

A main advantage against other classification models like KNN [7] is the consuming time in classification phase, while in KNN *n* comparisons are need to determine which class an unknown pattern belongs. With the proposed only one comparison is needed. In comparison with other associative memories models, the proposed model due the spherical neighborhood, works perfectly with patterns affected with additive, subtractive, and mixed noise.

The proposed method works perfectly when the classes are spherically separable. This procedure works with two classes but easily it can be extend to multi-class case.

5 Numerical Examples

In this section several illustrative examples are given. Due space limitations, only two examples are given. Numerical results, for the problem of finding an optimal spherical hull for a given set of patterns are presented. By simplicity, in order to clarify the results, a 2D and 3D Euclidean space for the geometric problem are used. Note that in these cases the spheres are circles and spheres respectively.

Example 1. Let the following two classes: $A = \{(1,1),(1,-1),(-1,-1),(-1,1)\}$ and $B = \{(2,2),(2,-2),(-2,-2),(-2,2)\}$; whose patterns are depicted in Fig. 1.

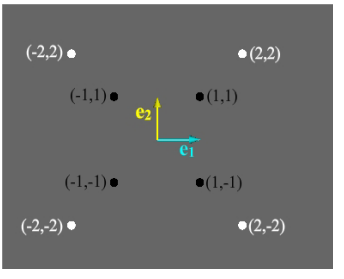


Fig. 1. Example 1, sets of patterns. Black points are patterns belong to class A and white points are patterns belong to class B.

From Fig. 1 it is clear that the classes are not linearly separable, but rather spherically separable. Next, the proposed method was used. The center of the optimal sphere S_1 is $(0,0)$ and its radius is 2.44. In Fig. 2 the resulting circle is plotted. In this case a right spherical hull for class A is found. It works as a separation surface between the two classes.

Let $a_i \in A$ and $b_i \in B$ for $i = 1,2,3,4$. The classification consists on applying an inner product between the obtained sphere and all the patterns. In this case $S_1 \cdot a_i = 2$ and $S_2 \cdot b_i = -1$ for $i = 1,2,3,4$. This is correct for the training patterns.

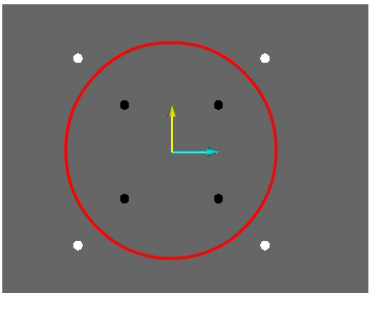


Fig. 2. Spherical hull for the class A. It functions as a separation surface.

Let the patterns $x_1 = [1,0], x_2 = [0,-1], x_3 = [0,0], x_4 = [0,1.5]$, where x_i is a distorted pattern of class A for $i = 1,2,3,4$. In these cases, an inner product between the patterns and the sphere will be applied:

$$S_1 \cdot x_1 = 2.5$$
$$S_1 \cdot x_2 = 2.5$$
$$S_1 \cdot x_3 = 3$$
$$S_1 \cdot x_4 = 1.87$$

For all the distorted patterns the classification is correct.

Example 2. Let the following two classes:

$A = \{(1,1,1),(1,-1,1),(-1,-1,-1),(-1,1,-1)\}$ and

$B = \{(2,2,-2),(2,-2,-2),(-2,-2,2),(-2,2,2)\}$; whose patterns are depicted in Fig. 3.

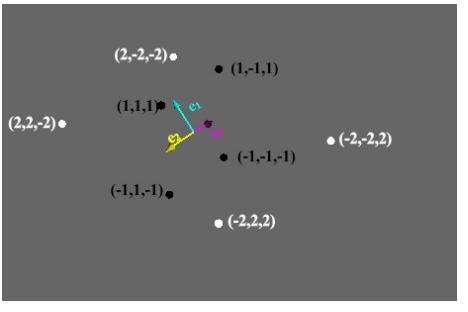


Fig. 3. Example 2, sets of patterns. Black points are patterns belong to class A and white points are patterns belong to class B.

From Fig. 3 it is clear that the classes are not linearly separable, but rather spherically separable. Next, the proposed method was used. The center of the optimal sphere S_2 is $\left(\frac{1}{3},0,\frac{1}{3}\right)$ and its radius is 3.19. In Fig. 4 the resulting sphere is plotted. The right spherical hull for class A works as a separation surface between the two classes.

Let $a_i \in A$ and $b_i \in B$ for $i = 1,2,3,4$. The classification consists on applying an inner product between the obtained sphere and all the patterns. In this case $S_2 \cdot a_i = 3.61$ and $S_2 \cdot b_i = -0.89$ for $i = 1,2,3,4$. This is correct for the training patterns.

Let the patterns $x_1 = [1,0,1], x_2 = [1,0,-1], x_3 = [0,0,0], x_4 = [0,1.5,-1.5]$, where x_i is a distorted pattern of class A for $i = 1,2,3,4$. In these cases, an inner product between the patterns and the sphere will be applied:

$$S_2 \cdot x_1 = 4.11$$
$$S_2 \cdot x_2 = 4.11$$
$$S_2 \cdot x_3 = 5.11$$
$$S_2 \cdot x_4 = 2.86$$

For all the distorted patterns the classification is correct.

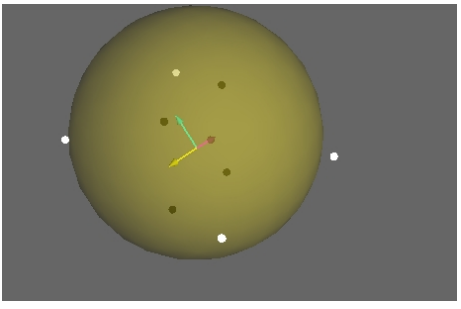


Fig. 4. Spherical hull for the class A. It functions as a separation surface.

6 Real Examples

Due space limitations only one real example will be given. Let the images shown in Fig. 5 the set of images to be classified. As mentioned before, the first step is to build respective spheres for each kind of image. For each class during training five different images were considered, these are called training images. For classification other ten different images were considered, these are called test images.

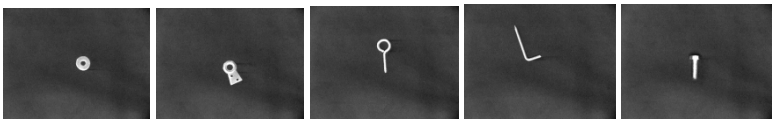


Fig. 5. Example images used to test the proposal

First a standard thresholder was used to convert the training images into binary images. An enhancement filter and was then used. An example of resulting image after the pre-processing is shown in Fig. 6. This procedure was done for each of the training images. In the last step the first four Hu invariants were obtained for each image.

Fig. 6. Examples of images after pre-processing

With these Hu invariants as 4-element vectors, the respective spherical neighborhoods were generated. In this case five spherical neighborhoods were generated. First test consisted on classifying the training images using their respective Hu invariants and equation (15) for each sphere. In this case a 100% of recovering was obtained in all cases; all the training images were correctly classified.

The second test consisted on classifying the test images using their respective Hu invariants and equation (15) for each sphere. The classification results per class are shown in Table 1.

Table 1. Results of second test

Kind of Image	Num. of Images	Percent of Recovering
Gasket	10	100%
Duck Tail	10	100%
Eyebolt	10	90%
Pitons	10	90%
Screw	10	100%

As shown, in this case only two classes were not correctly classified. A third test was performed. In this case the images used for testing were altered with mixed noise. Examples of the corrupted images are shown in Figure 7.

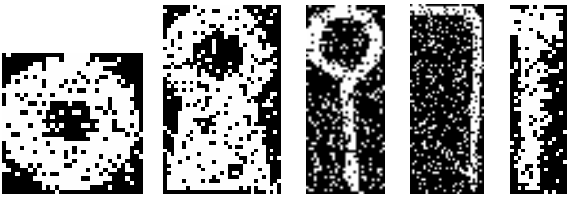


Fig. 7. Examples of noised images used to test the proposal.

The images were altered with 20% of salt-pepper noise. The same pre-processing was applied to these corrupted images. The classification results are shown in Table 2.

As shown, in this case the results not good as expected, due to some patterns fall out of the sphere. A new algorithm to adjust the spheres with a maximum radius would be thus necessary.

Table 2. Results of third test

Kind of Image	Num. of Images	Percent of Recovering
Gasket	10	50%
Duck Tail	10	70%
Eyebolt	10	60%
Pitons	10	40%
Screw	10	60%

7 Conclusions and Future Work

Geometric Algebra allows to model situations and to enunciate problems in terms of high level symbolic expressions. Nevertheless, it is possible to achieve an implementation working in the elementary coordinate system. Of course that in some cases, it is possible to find a solution by purely handling symbolically the expressions, but this is unusual in the case of realistic problems.

In this work a new classification model based on Conformal Geometric Algebra principles was described. Spherical neighborhoods as decision regions were used.

Optimization techniques were used to construct an optimal spherical neighborhood. A method to transform the problem of finding an optimal sphere to a quadratic programming was described. The method functions perfectly when the classes are spherically separable and with two classes. It can be easily extended to other multiclass situations.

For classification purposes, an inner product between the unclassified pattern and the GAM is applied. Then a minimum function is used to obtain the index class. The result is a class index.

Numerical and real examples were given to demonstrate the potential of the proposal. As shown, the method can operate with non-linearly-separable patterns. The proposed model can also efficiently cope with distorted patterns.

Nowadays, we are finding the formal conditions under which the proposed model can work. We are also interested to test it in more realistic situations. Furthermore, we are working in a way to maximize sphere's radius to enhance test with noised patterns.

Acknowledgements. This work has been supported by the National Polytechnic Institute of Mexico (SIP-IPN), under grant 20082921 and 20082948, and by the Mexican Science and Technology National Council (CONACyT), under grant 46805.

References

1. Hildebrand, D., Fontijne, D., Perwass, C., Dorst, L.: Geometric algebra and its application to computer graphics. In: EUROGRAPHICS 2004 Tutorial 3. Eurographics Association, Aire-la-Ville (2004)
2. Clifford, W.K.: Applications of Grassmann's Extensive Algebra. American Journal of Mathematics 1(4), 350–358 (1878)
3. Hestenes, D., Sobczyk, G.: Clifford Algebra to Geometric calculus, A Unified language for Mathematics and Physics. D. Reidel Publishing Company (1984)
4. Dorst, L., Fontijne, D.: 3D Euclidean geometry through conformal geometric algebra (a GAViewer tutorial) (2005), http://www.science.uva.nl/ga/tutorials/CGA/
5. Hildebrand, D.: Geometric computing in Computer graphics using Conformal Geometric Algebra, Interactive Graphics Systems Group, TU Darmstadt, Germany (2005)

6. Perwass, C., Hildebrand, D.: Aspects of geometric algebra in Euclidean, projective and conformal space. Technical Report Number 0310, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik (2003)
7. Cover, T., Hart, P.: Nearest Neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
8. Barrón, R., et al.: New Improved Algorithm for the Training of a Morphological Associative Memory. Research in Computing Science 21, 49–59 (2006)
9. Barron, R., Cruz, B., Laguna, G.: Conformal Geometric Algebra for Spherical Convex Hull Optimization. In: AGACSE 2008 (accepted for publication, 2008)

Automatic Classification of Three-Phase Flow Patterns of Heavy Oil in a Horizontal Pipe Using Support Vector Machines

Adriane B.S. Serapião[1], Antonio C. Bannwart[2], Fabíola Pacheco[2], and José R.P. Mendes[2]

[1] UNESP /IGCE/DEMAC – C.P. 178 – CEP 13506-900 – Rio Claro (SP), Brazil
adriane@rc.unesp.br
[2] UNICAMP/FEM/DEP – C.P. 6122 – CEP 13081-970 – Campinas (SP), Brazil
{bannwart,fabiola,jricardo}@dep.fem.unicamp.br

Abstract. The pipe flow of a viscous-oil-gas-water mixture such as that involved in heavy oil production is a rather complex thermo-fluid dynamical problem. Considering the complexity of three-phase flow, it is of fundamental importance the introduction of a flow pattern classification tool to obtain useful information about the flow structure. Flow patterns are important because they indicate the degree of mixing during flow and the spatial distribution of phases. In particular, the pressure drop and temperature evolution along the pipe is highly dependent on the spatial configuration of the phases. In this work we investigate the three-phase water-assisted flow patterns, i.e. those configurations where water is injected in order to reduce friction caused by the viscous oil. Phase flow rates and pressure drop data from previous laboratory experiments in a horizontal pipe are used for flow pattern identification by means of the 'support vector machine' technique (SVM).

1 Introduction

Three-phase pipe flow of a viscous oil-water-gas mixture is one of the biggest challenges in petroleum production, in view of magnitude of heavy oil reserves worldwide (trillions of barrels). The complexity of multiphase flow is such that theoretical models for fundamental quantities such as pressure and temperature variations along the pipe can be developed only for very few exceptional phase configurations. Those quantities govern phase equilibria as well as thermo-physical properties of each phase and many undesirable phenomena such as asphaltene/paraffin depositions, hydrate formation, emulsion generation, etc.

Temperature and pressure drop control in under deepwater pipelines become crucial when viscous oils are involved, in view of the low temperature at the sea bottom and its potentially dramatic effect on the oil viscosity increase. Pressure drop control techniques, as focused in this work, are aimed at friction reduction by lowering the effective viscosity of the fluid being produced. One of the most important friction reduction techniques is provided by the continuous injection of water (or other low

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 284–294, 2008.

viscosity liquid) in the pipeline, so as to avoid direct oil-wall contact. Flow patterns of this type are known as 'water-assisted flow' or 'water-continuous flow'.

However, produced oil is usually accompanied with dissolved gas, which releases from the oil phase as far as the pressure falls below the bubble point pressure, thus creating a further free gas phase. The resulting three-phase mixture presents itself in several spatial distributions of the phases (i.e. flow patterns) within the pipeline, each one with a key effect on the pressure drop. Due to the inherent complexity of the three-phase flow structure and the flow-pattern-dependent nature of pressure drop, investigations can only be performed on the basis of laboratory experiments. Thus of the development of an objective flow pattern identification scheme is of fundamental importance to obtain useful information about the flow nature and for control purposes as well.

In a previous paper [1], the three-phase water-assisted flow of heavy crude oil with gas (air) in a horizontal pipe was investigated at a low pressure laboratory set up. For each trio of flow rates, flow patterns were identified by means of direct visualization (with the help of movie recording) and total pressure drop was measured for comparison with existing correlations.

In the present work, we investigate the possibility of predicting the horizontal three-phase flow patterns described in [1] from the correspondent flow rates and pressure gradient data. This issue becomes relevant when direct visualization or tomography of the flow is not possible or unfeasible, such as in a real heavy oil production pipeline. For that purpose we use a 'support vector machine' tool (SVM) which is well-known in the field of artificial intelligence [2]. In our previous work [3], we investigated the use of SVM for vertical pipe flow patterns, with interesting results.

We start this paper by summarizing, in Section 2, the experimental settings and the observed horizontal flow patterns. In Section 3 the underlying support vector machine methodology used for the classification task is briefly explained. Results and discussions are given in Section 4 followed by some concluding remarks in the last Section.

2 Experiments in Three-Phase Water-Assisted Flow of Heavy Oil with Free Gas

Many onshore and offshore production flow lines connecting the well heads to the production and storage facilities have horizontal segments or are entirely horizontal. In the case of heavy oil production, water can be injected in the pipeline for the purpose of friction reduction.

In the following we summarize the test conditions and the main results obtained in the laboratory experiments described in [1]. The test section consisted of a 2.84 cm i.d., 2.5 m long horizontal glass tubing for the three-phase flow. The oil flow rate was measured with a Coriolis mass flow meter, whereas water and air flow rate measurements were provided by rotameters. Pressure data in the test section were measured with differential and absolute pressure transducers connected to a data acquisition system.

The oil utilized was a blend of crude dead oil with a viscosity of μ_o = 5,040 mPa.s and a density of ρ_o = 971 kg/m^3 at 25^{o}C. The oil phase collected from the three-phase separator was observed to be a w/o emulsion (i.e. oil with dispersed water

micro-droplets). This was due to the continuous recirculation of both oil and water to the separator, whereas air was vented. Water contained in the separator was tap water and air was provided by an existing group of compressors.

The experiments consisted of simultaneously flowing water, crude oil and air at several flow rate combinations. For each set a video footage of the established flow pattern was taken with a high-speed camera (1000 frames/s) and pressure data were collected. The experimental superficial velocities varied within the following ranges:

- oil: $0.02 < J_o < 1.2$ m/s
- air: $0.04 < J_g < 9$ m/s
- water: $0.04 < J_w < 0.5$ m/s

The experiments took place at ambient temperature and near atmospheric pressure. In all runs, water was always injected first (in order to make sure that it would be the continuous phase), followed by oil and air. The glass pipe was never observed to be fouled (hydrophilic behavior).

The classification of the three-phase flow patterns in the horizontal pipe was done by individually classifying the air-water and oil-water flows. The following nomenclature was used for each step: **B**- Dispersed Bubbles, **A**- Annular, **S**- Stratified and **I**- Intermittent. After classifying both two-phase flows, these symbols were combined to form the three-phase horizontal flow designation. The nine heavy oil-water-air flow patterns obtained are illustrated in Figure 1.

A description of the nine horizontal flow patterns is as follows:

- **Bg-Bo: Bubbly gas – Bubbly oil** (Fig.1.a)**.** Spherical oil bubbles were observed in this pattern. Dispersed through the pipe cross section, these bubbles follow the water velocity profile (higher velocities near the center of the tube and lower velocities near the tubing walls). The gas phase is also consisted of bubbles, although located at the top of the pipe cross section, due to the lower density of the gas.
- **Bg-Ao: Bubbly gas – Annular oil** (Fig.1.b)**.** In this case the oil phase forms a continuous body in the center of the tube with water flowing near the walls. The gas flows in the same pattern described in the previous pattern; it also appeared between the oil interfacial waves for low water flow rates.
- **Bg-Io: Bubbly gas – Intermittent Oil** (Fig.1.c)**.** In this pattern, the oil phase presented itself as bubbles of larger sizes that coalesced to form elongated bubbles, flowing through the upper portion of the pipe.
- **Bg-So: Gas Bubbles – Stratified Oil** (Fig.1.d)**.** The stratified oil flow, as in the previous annular flow, consists of a continuous phase, although in this case the oil flowed very near the top due to gravity segregation. However, a thin water film between the oil and the top wall was observed which explains its high velocity. The existence of this film can be attributed to the wet ability of the glass, which is hydrophilic. The gas phase flowed as bubbles between the oil phase and the top wall.
- **Ig-Bo: Intermittent Gas – Oil Bubbles** (Fig.1.e)**.** The increase in the gas flow rate caused an increase in the size of its bubbles, many times occupying almost the whole cross section, except for a small liquid fraction. These elongated gas bubbles caused strong agitation in the liquid phases, composed by a water continuous phase with dispersed oil bubbles.

- **Ig-Ao: Intermittent Gas – Annular Oil** (Fig.1.f)**.** The liquids configuration in this pattern consisted of a continuous oil phase lubricated by a surrounding water film, even though the passage of a faster elongated gas bubble.
- **Ig-Io: Intermittent Gas – Intermittent Oil** (Fig.1.g)**.** The liquid phases in this pattern consisted of a continuous water phase which kept in contact with the pipe wall, and oil as elongated bubble. At higher air flow rates, the water phase displayed small dispersed oil droplets, due to the strong agitation caused by the gas phase.
- **Sg-Bo: Stratified Gas – Oil Bubbles** (Fig.1.h)**.** This flow pattern consisted of a continuous gas phase stratified at the top of the pipe. The oil phase stabilized as spherical bubbles flowing at the top of the aqueous phase, which was continuous and, in some cases, presented a wavy interface.
- **Sg-So: Stratified Gas – Stratified Oil** (Fig.1.i)**.** This flow pattern exhibited the gravity segregation of the three continuous phases and occurred for low flow rates of oil, water and gas. The liquid phases presented interfacial waves that did not touch the top wall.

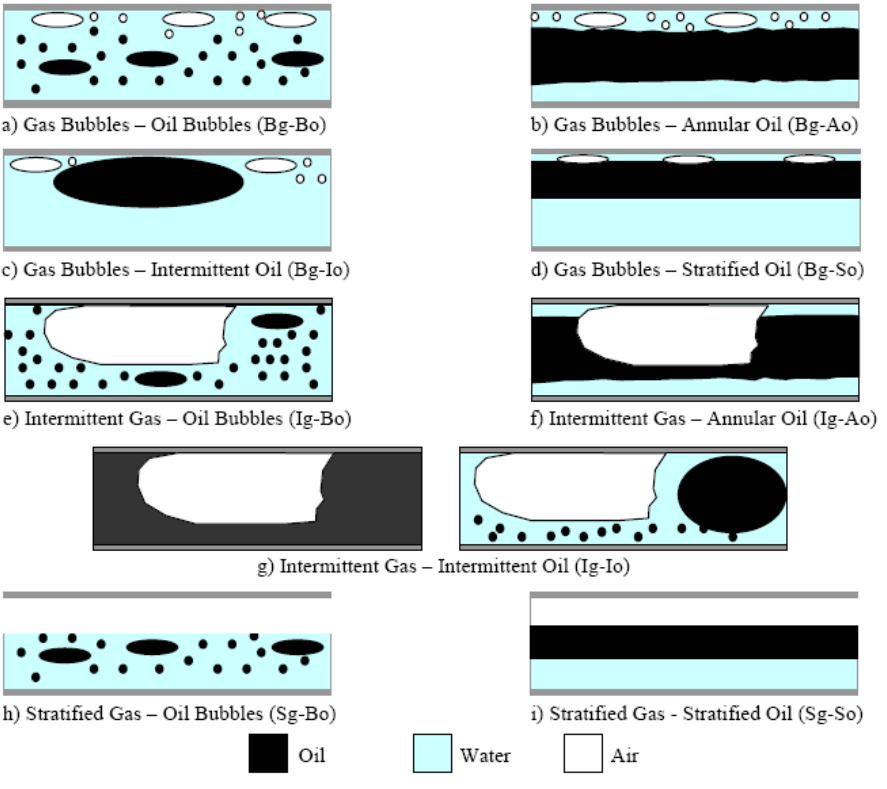


Fig. 1. Three-phase patterns for horizontal water-assisted flow of heavy oil in the presence of a free gas phase

3 Support Vector Machines

SVMs have been successfully applied to solve a large number of classification tasks [4]. They are based on the principles of structural risk minimization (SRM) and hence have good generalization ability. In the simplest and linear form, SVM is the optimal hyperplane that separates a set of positive samples from a set of negative samples, with an margin defined by the distance between the hyperplanes supporting the nearest positive and negative samples and at the same time reducing the empirical risk.

The power of SVMs lies in their ability to transform data to a higher dimension and construct a linear binary classifier in the higher dimension. The construction of the linear decision boundary is done implicitly and hence the sparseness of the data in the higher dimension is not an issue [2].

For the two-class pattern classification problem, SVM finds a hypersurface $y(x)$ decision, where the vector X belongs to the space of samples [5]. Analyzing a set of N data points $\{x_i, y_i\}$, where $x_i \in \Re^n$ is the i-th input data, and $y_i \in \{-1, +1\}$ is the label of the data. The SVM approach aims at finding a classifier of form [4]:

$$y(x) = sign\left[\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right] \tag{1}$$

where α_i are positive real constants and b is a real constant, in general, $\phi(x)$ is the nonlinear map from the original space to the high dimensional space, and $K(x_i,x) = \langle \phi(x_i), \phi(x) \rangle$, $\langle \bullet,\bullet \rangle$ is an inner product called the kernel function.

In the high dimensional space, we assume the data can be separated by a linear hyperplane, this will cause:

$$\begin{cases} w^T \phi(x_i) + b \geq 1, & \text{if } y_i = +1 \\ w^T \phi(x_i) + b \leq -1, & \text{if } y_i = -1 \end{cases} \tag{2}$$

which is equivalent to:

$$y_i[w^T \phi(x_i) + b] \geq 1, \quad i = 1,...,N \tag{3}$$

If the separating hyperplane does not exist, a so called slack variable ξ_i is introduced such that:

$$\begin{cases} y_i[w^T \phi(x_i) + b] \geq 1 - \xi_i, & i = 1,...,N \\ \xi_i \geq 0 & i = 1,...,N \end{cases} \tag{4}$$

where: w is the vector of coefficients, b a constant and ξi are parameters for handling non-separable data (inputs). The index i labels the N training cases. Note that $y \in \pm 1$ is the class labels and x_i is the independent variables. The kernel ϕ is used to transform data from the input (independent) to the feature space.

In order to restrict the expressiveness of the hypothesis space, the SVM searches for the simplest solution that classifies the data correctly. The learning problem is hence reformulated according to the structural risk minimization principle. The risk bound ($J_1(w,\xi)$) is minimized by the following minimization problem, which is equivalent to maximizing the distance, normal to the hyperplane, between the convex hulls of the two classes (margin):

$$\min_{w,\zeta} J_1(w,\xi) = \frac{1}{2} w^T w + c\sum_{i=1}^{N} \xi_i \tag{5}$$

subject to the constraints (4), where c is the capacity constant.

This problem is tractable but in order to proceed to the non-separable and non-linear cases, it is useful to consider the dual problem as outlined below:

$$L_1(w,b,\xi,\alpha,\beta) = J_1(w,\xi) - \sum_{i=1}^{N} \alpha_i \left\{ y_i \left[w^T \phi(x_i) + b \right] - 1 + \xi_i \right\} - \sum_{i=1}^{N} \beta_i \xi_i \tag{6}$$

where $\alpha_i \geq 0$, $\beta_i \geq 0$ $(i = 1, \ldots, N)$ are the Lagrangian multipliers of (4). The solution to this quadratic programming problem is given by maximizing L with respect to α,β and minimizing with respect to w,b,ξ:

$$\max_{\alpha,\beta} \min_{w,b,\xi} L_1(w,b,\xi,\alpha,\beta) \tag{7}$$

Differentiating with respect to w, b and ξ and setting the derivatives equal to zero yields:

$$\frac{\partial L_1}{\partial w} = w - \sum_{i=1}^{l} \alpha_i y_i w_i = 0 \tag{8}$$

and

$$\frac{\partial L_1}{\partial b} = -\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{9}$$

and

$$\frac{\partial L_1}{\partial \xi_i} = 0, \text{ with } 0 \leq \alpha_i \leq c,\ i = 1,\ldots, N \tag{10}$$

So that the optimal solution is given by (4) with weight vector:

$$w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \tag{11}$$

Substitute (8), (9) and (10) for (6), we will get the following quadratic programming problem:

$$\max_{\alpha} Q_1(\alpha) = -\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum_{i=1}^{N} \alpha_i \tag{12}$$

where $K(xi,xj) = \langle \phi(xi), \phi(xj) \rangle$, is called the kernel function.

It is possible to use different types of functions as sigmoid, gaussian radial basis (RBF), polynomial and linear functions in order to implement the kernel function (13):

$$\phi = \begin{Bmatrix} x_i * x_j & linear \\ (\gamma . x_i x_j + coefficien\, t)^{\deg ree} & polynomial \\ \exp(-\gamma \left| x_i - x_j \right|^2) & RBF \\ \tanh(\gamma . x_i x_j + coefficien\, t) & sigmoid \end{Bmatrix} \tag{13}$$

Solving this quadratic programming problem will place the hyperplane in the high dimensional space and hence the classifier in the original space as in (1). To construct an optimal hyperplane, SVM uses an iterative training algorithm, which is used to minimize an error function (5). This SVM model is known as *Classification SVM Type 1 (also known as C-SVM classification).*

At first, SVM was originally designed to solve binary classification problem. Later they could also be extended to solve multicategory problems by the combination of binary SVM classifiers, where the data set is decomposed to several binary problems. In this work we use the one-against-one strategy. This approach trains a binary SVM for any two classes of data and obtains a decision function. Thus, for a C-class problem, there are $C(C-1)/2$ decision functions. In the prediction stage, a voting strategy is used where the testing point is designated to be in a class with the maximum number of votes.

4 Results and Discussion

We used multi-class SVM Type 1 as training algorithm for classification, considering the "one-against-one" method for decomposing multi-class problem in binary subproblems. The linear, polynomial and gaussian RBF functions were tested as the mapping function (kernel) for the classification system. The Matlab®1 v.7.4 was used to create the SVM model employed in this work.

The SVM classifier is modeled using the four inputs, representing the independent variables (oil flowrate, water flowrate, gas flowrate and pressure gradient) and the output is one of the nine target class previously classified by the expert (Bg-Bo, Bg-Ao, Bg-Bo, Bg-So, Ig-Bo, Ig-Ao, Ig-Io, Sg-Bo and Sb-So), representing each flow pattern, according to Figure 2.

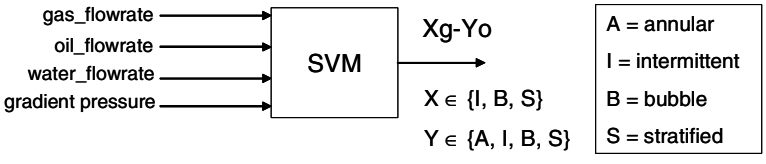


Fig. 2. SVM classification system

Experimental records of three-phase flow of heavy oil with gas and water in a *horizontal pipe* data set consisting of 119 samples were used for the training and evaluation of the implemented SVM classifier. When training SVM network the whole data set was randomly separated into two subsets: 75% as training subsets (89 samples) and 25% as testing subsets (30 samples) after training. The test subset is used to check the ability of generalization of the classification system, where characteristic signals for each of the flow regimes are presented to the trained SVM, in order to verify whether the expected class is output by the SVM classifier. The distribution of samples for each class (pattern) in the training and test datasets is shown in the Table 1.

[1] Product of *MathWorks, Inc*.

The performance of the SVM classifier employed to identify the flow patterns was assessed through comparisons between original and estimated outputs taken from the data subsets used both in training and in testing procedure samples for three different types of kernels. In the linear SVM the capacity constant (c) was set to infinite (∞). The implementation of the polynomial kernel has considered a 5 degree polynomial with $c = 1$, $\gamma = 2$ and coefficient = 100. The RBF kernel has adopted $c = 500$ and $\gamma = 1$. All the experiments were obtained on a Centrino Duo PC (CPU 1.83 GHZ, RAM 2GB). After training, the SVM has found 41, 42 and 71 support vectors to construct the classes' separation hypersurface for linear, polynomial and RBF kernels, respectively.

Table 1 shows the results comparing the statistical performance of SVM classifier using the three different kernels for both training and test datasets for a flow in a horizontal pipe. The results of our experiments can be summarized as follows. Concerning classification accuracy, the polynomial kernel obtained results somewhat better than linear and RBF kernels in both data sets, whereas linear and RBF kernels obtained exactly the same performance during the training and testing phases. Accuracy values are in percentage of success. The polynomial kernel correctly classified all samples of the training set, with 100% of recognition score. For the test set, just one sample was misclassified, reaching a reliability of 96.67%. For the linear kernel, three samples presented a false classification in the training set and two samples indicated incorrect identification in the test set (respectively, 96.63 and 93.33% of correct samples). Although the RBF kernel has had no error for classification during the training phase, this kernel has presented the worst performance among them, with three misclassifications in the test set (90% of success).

Comparing the misclassified patterns in all kernels, we have verified that, using the polynomial kernel, a Sg-Bo sample was recognized as a Bg-So pattern in the test set. Using the linear kernel, for the training set two samples of Ig-Bo pattern were placed as belonging to the Ig-Io class, whereas one sample of the Ig-Io pattern was labeled as Ig-Bo. In the test set, Ig-Io was also wrongly identified as Ig-Bo pattern and Sg-Bo as Bg-So. With RBF kernel, one sample of both Ig-Bo and Ig-Io was ranked as Sg-Bo class in the test set, whilst one Sg-Bo sample was assigned as Bg-So class.

Table 1. SVM Statistical for horizontal pipe

Pattern	Training set				Test set			
	Nb. of samples	Misclassifications			Nb. of samples	Misclassifications		
		Linear	Polynomial	RBF		Linear	Polynomial	RBF
Bg-Ao	11	0	0	0	2	0	0	0
Bg-Io	2	0	0	0	2	0	0	0
Bg-Bo	5	0	0	0	2	0	0	0
Bg-So	4	0	0	0	2	0	0	0
Ig-Ao	8	0	0	0	2	0	0	0
Ig-Bo	23	2	0	0	7	0	0	1
Ig-Io	30	1	0	0	10	1	0	1
Sg-Bo	5	0	0	0	2	1	1	1
Sg-So	1	0	0	0	1	0	0	0
Total	*89*	*3*	*0*	*0*	*30*	*2*	*1*	*3*
Accuracy rate		**96.63%**	**100%**	**100%**		**93.33%**	**96.67%**	**90%**

An incorrect classification of the Sg-Bo as Bg-So was found for all kernels. This mistake can be explained because of the resemblance between these patterns, as showed in the Fig. 1d and 1i. In analogous way, we can observe likelihood between Ig-Io and Ig-Bo patterns, in conformity with Fig. 1e and 1g.

Still according to Table 1, all samples of Bg-Yo, Ig-Ao and Sg-So classes, where Y represents (A)nnular, (I)ntermittent, (S)tratified or (B)ubble pattern, obtained a correct classification using any kernel for the SVM classifier for both training or test dataset.

The results obtained using SVM for horizontal flow pattern detection were compared with two models of neural networks: perceptron multi-layer (MLP) with backpropagation (BP) and radial basis function (RBF) network [7]. On MLP neural network, four neurons were used in input layer, representing the independent variables (oil, gas and water flow rates, and gradient pressure), 13 neurons were employed in hidden layer and nine neurons were used in output layer, representing each horizontal flow pattern. The amount of neurons in the hidden layer was found by experimentation, after assessing which configuration would yield the least global training error. The activation functions used at each level were sigmoid tangent for both hidden and output layer neurons. The supervised training algorithm used was the Backpropagation and the training consisted of 5000 epochs. The learning rate was set to 0.4 and the momentum term was equal to 0.2.

In the RBF network [8], the input layer was composed by one neuron for each predictor variable (four inputs). The hidden layer had a variable number of neurons, where the optimal number was determined during the training process. Each neuron consisted of a radial basis function centered on a point with as many dimensions as there were predictor variables. The hidden neurons have used symmetric multivariate gaussian functions with coefficient of spread (γ) equal to 0.5. Gaussians are fit to the data from each cluster. In the output layer, one neuron was for each target class (nine flow patterns), with a separate set of weights and summation unit. The value output for a class is the probability that the case being evaluated has that category. The k-means clustering was used to find cluster centers which were then used as the centers for the radial based functions. Two clusters ($k = 2$) were used in the k-means algorithm. All numeric attributes were standardized to zero mean and unit variance for training.

We have also investigated the performance of all these techniques with vertical pipes for the pattern recognition task, as initiated in our previous work [3]. Table 2 shows the accuracy rate for each flow classification method for both pipes, considering the training, test and full datasets. This last one is the original set, combining training and test datasets.

MLP-BP neural network obtained a best result than RBF neural network for flow pattern identification in horizontal pipe, while the RBF network reached a higher performance in the vertical pipe in contrast with the MLP-BP network, as reported in Table 2. Comparing neural networks and SVMs classifiers, we can notice that, in general, the SVM carried out more thoroughly blanket flow pattern detection, especially on account of the test subset. Polynomial SVM outperformed all others implemented classifier systems for both horizontal and vertical pipes.

It is important to mention that there were identified only six flow patterns with vertical upward flow [9], against nine patterns of the horizontal flow [1]. Although the

number of vertical patterns is lower, the pattern recognition task for vertical flow has shown to be more complex that for the horizontal flow. We have used the same input parameters (gradient pressure and flow rates) for developing the classifier systems for both types of pipe. Results show that the SVM is capable of correctly identifying the flow regimes.

Table 2. Comparing accuracy of flow classifiers in horizontal and vertical pipes

	Horizontal pipe			Vertical pipe		
Classifier	***Training set***	***Test set***	***Full dataset***	***Training set***	***Test set***	***Full dataset***
MLP-BP NN	98.88%	86.67%	95.72%	89.89%	60.0%	82.35%
RBF NN	100%	80.0%	94.96%	91.01%	66.67%	84.87%
Linear SVM	96.63%	93.33%	95.80%	85.39%	70.0%	81.51%
Polynomial SVM	100%	96.67%	99.16%	100%	73.33%	93.28%
RBF SVM	100%	90.0%	97.48%	100%	70.0%	92.44%

Computational time required to solve the problem classification is governed primarily by the time required for training. The spent time for reaching the solution was very short, around 0.5 second for SVM classifiers. The time scale for training with neural networks was in the order of seconds or minutes.

5 Conclusion

The experimental results show that the proposed SVM for the horizontal flow pattern identification task achieved high predictive accuracy. SVM has showed to be a more powerful tool than neural networks for horizontal and vertical flow identification as well.

Comparing the horizontal flow pattern identification with the vertical one, it is possible to note that the horizontal flow has a recognition score higher than the vertical flow pattern. Intermittent pattern in horizontal pipes does not cause so hard misclassification as in vertical pipe, as supposed in [3]. This shows that horizontal flow patterns are easier identified than vertical ones. Furthermore, the polynomial kernel appeared to be slightly more suitable one for this work, since such model got the best performance on predicting values of non-training data, either for vertical or horizontal flow pattern identification. Thus, taking this model as the most promissory is, exclusively, due to the fact of its better generalizations for patterns not known *a priori*. In general, the reliability of the results assures the applicability of SVM to solve the problem presented herein. The results obtained thus far promise to be profitable for surveillance, acquisition knowledge and operational control in heavy oil processes.

The application of methodologies of Artificial Intelligence in the petroleum industry is encouraged to solve complex automation problems in order to increase oil production and to reduce costs, without compromising safety.

Finally, the authors would like to state that this is a pioneer study in the application of SVM to predict three-phase flow patterns. There is a strong interest in continuing research in this field, aiming at better accuracy rates results for wider datasets. In order to achieve this goal, others data mining methods will be studied and applied to

the problem treated herein. We also plan to make further experiments to understand the influence of system's parameters on the performance of a flow pattern classifier, so that, we can set others combinations of measured or calculated variables (*i.e.*, absolute pressure, temperature, BSW) in term of the same identification problem.

Acknowledgements

The authors would like to thank the Brazilian Research Council (CNPq) and Fundunesp for the partial support for this work.

References

1. Trevisan, F., Bannwart, A.C.: Three-phase flow patterns and pressure drop for heavy crude oil-water-gas horizontal flow, paper IHTC13-1322. In: Proceedings of the 13th International Heat Transfer Conference - IHTC, Sydney, Australia (2006)
2. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
3. Pacheco, F., Bannwart, A.C., Mendes, J.R.P., Serapião, A.B.S.: Support Vector Machines for Identification of Three-Phase Flow Patterns of Heavy Oil in Vertical Pipes. Brazilian Journal of Petroleum and Gas 1(2), 95–103 (2007)
4. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, Chichester (1998)
5. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 1–47 (1998)
6. Hsu, C.-W., Lin, C.-J.: A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Networks 13(2), 415–425 (2002)
7. Haykin, S.: Neural Networks: A comprehensive foundation, 2nd edn. Prentice Hall, Englewood Cliffs (1999)
8. Poggio, F.: Regularization theory, radial basis functions and networks. In: From Statistics to Neural Networks: Theory and Pattern Recognition Applications. NATO ASI Series, vol. 136, pp. 83–104 (1994)
9. Bannwart, A.C., Vieira, F.F., Carvalho, C.H.M., Oliveira, A.P.: Water-assisted flow of heavy oil and gas in a vertical pipe, paper PS2005-SPE-97875-PP, 2005. In: SPE International Thermal Operations and Heavy Oil Symposium - ITOHOS, Calgary, Alberta, Canada (2005)

Wavelets Based on Atomic Function Used in Detection and Classification of Masses in Mammography

Cristina Juarez-Landin[1,2], Volodymyr Ponomaryov[1], Jose Luis Sanchez-Ramirez[1], Magally Martinez-Reyes[2], and Victor Kravchenko[3]

[1] National Polytechnic Institute, ESIME-Culhucan, Av. Santa Ana, 1000, Col. San Fco. Culhuacan, 04430, Mexico, D. F. Mexico
[2] Autonomous University of State of Mexico, CU Valle of Chalco, Av. Hermenegildo Galeana, 3, Col. Ma. Isabel, 56615, Edo. de Mexico, Mexico
[3] Institute of Radio Engineering and Electronics of Russian Academy of Sciences, Moscow
cjlandin@yahoo.com.mx, vponomar@mail.ru, jluissar@yahoo.com.mx, kvf@pochta.ru

Abstract. Mammography is considered the most effective method for early detection of the breast cancer. However, it is difficult for radiologists to detect microcalcification (MC) clusters and camouflages masses. The mammograms (MG) images were decomposed into several subimages using Wavelet transform (WT) based on classical and novel class Wavelets using atomic function for reducing the volume of data in the classification stage. Various regions of interest (ROIs) in the MG images were selected where input data for multilayer artificial neural network (ANN) type classifier are formed applying the WT. We used different patterns to classify the normal, MC, spiculated and circumscribed masses ROIs. The detection performance has been evaluated on MG images from the Mammographic Image Analysis Society (MIAS) database. The proposed classification scheme was shown good performance in detecting the MC clusters and masses with acceptable classification.

Keywords: Classification, Neural Networks, Mammography, Microcalcification, Wavelet Transform, Atomic Function.

1 Introduction

A variety of classical methodologies from signal processing and pattern recognition areas have been implemented and tested for diverse medical applications [1], [2], [3]. Here, we discuss different computer vision techniques that are employed for computer-aided detection (CAD) of breast cancer. Such the CAD systems in MG, and especially MC detection and diagnosis could provide remarkable support as a "second opinion" tool, improving the effectiveness of the decision-making procedure. Below, the techniques of the image processing and the image pattern recognition are presented in application to MG images.

Mammography is a medical image modality that may be digital or not digital. The main MG signs of breast cancer are clustered MC and masses. Because of the nonspecific features of malignant lesions, MG interpretation is a very challenging task for

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 295–304, 2008.

radiologists. The sensitivity and specificity of MG for detecting a lesion and differentiating the lesion as malignant or benign should be improved.

By now, many approaches have been proposed for the automatic detection and recognition of clusters of MCs and masses [4]. Usually, the first step is the preprocessing of an image, aimed to improve the quality of the MG in terms of contrast and noise. After that, a MC detection phase is carried out to localize the clusters on the MG. The final step is the classification of the cluster or mass as malignant or benign. Rashed *et al.* [5] proposed a method based in multiresolution analysis of the MG and the pattern classification that was carried out for half Euclidean distance measurements.

In difference with referred woks, we use in this study novel class of Wavelets that are based on atomic functions, which have some advantages that permit to improve the classification and detection of the MG patterns. Additionally, we propose to construct and evaluate a supervised classifier for mammograms applying a wavelet transform decomposition. Another important classification is physical nature of the lesions, such as benign or malign. The distinction between these two classes is ill-defined in terms of the image analysis. So, usually a physician asks for additional analysis that includes other tests to characterize the tumor as benign or malign. In the case of automated computer classification, we propose that it is not necessary to keep and use all of the coefficients for classification, and in fact we tested portions of the approximation coefficients applying thresholding strategies to suit a pattern recognition problem. So, the biggest approximation coefficients of the decomposed image are used as a vector of the elements of the class to be classified.

Present proposal is consists of constructing a classification scheme and evaluation an ANN supervised classifier for MGs using a decomposition based on classical and an atomic function *eup(x)* Wavelets. The experiments performed here show that sufficiently successful classification can be achieved, when we investigate two different problems: (1) classification between normal, MCs, radial or spiculated masses, and circumscribed masses areas; (2) classification between normal, benign, and malign areas.

Here, we present computerized scheme for the classification of clustered microcalcifications (MC), radial or spiculated masses (SM), and circumscribed masses (CM) in digital MGs by using region-based segmentation, WT, and ANN techniques.

In this work, multilayer perceptron (MLP) structure is used with globally supervised learning rules.

It was found in the tests carried out previously that better results were obtained when classification is realized applying the Daubechies Wavelet for MG images [6]. Based on this fact we investigate classification performance using this Wavelet function with recently proposed by one of author [7], novel Wavelets based on Atomic Functions (WAF).

2 Method and Techniques

Fig. 1 shows the flow of the proposed scheme that consists of stages where the spot-like characteristics in the original X-ray image are enhanced before realizing border detection and classification for ANN.

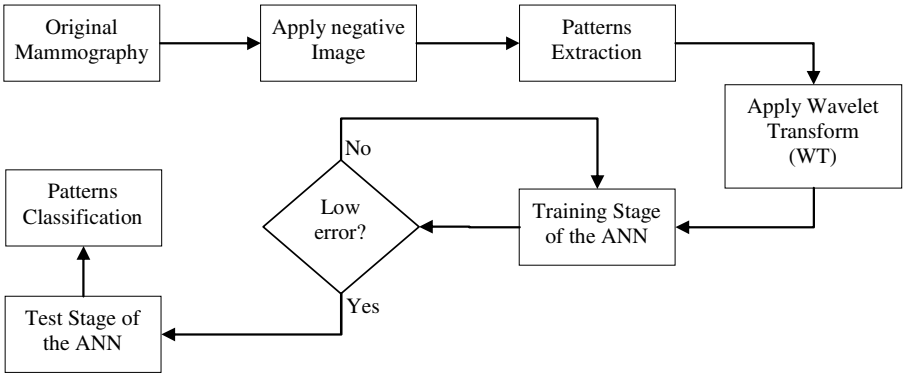


Fig. 1. Proposed classification scheme

2.1 Data Sources

Because it is difficult to access real medical images for experimentation due to privacy issue, the data collection was taken from the MIAS database in the experiments realized here. MIAS database consists of 322 images classified in three categories: normal, benign and malign, which are considered abnormal [8]. Additionally, the abnormal cases are further divided into six categories: circumscribed masses, spiculated masses, microcalcifications, ill-defined masses, architectural distortion and asymmetry. All the images are digitized at resolution of 1024x1024 pixels and 8-bit accuracy in gray level. They also include the locations of any abnormalities that may be present. The existing data in the collection consists of the location of the abnormality (like the center of a circle surrounding the tumor), its radius, breast position (left or right), type of breast tissues (fatty, fatty-glandular and dense) and tumor type if exists (benign or malign). It has been realized classification stages employing two different parts of MIAS database, one in training phase, another during test phase.

2.2 Patterns of the MG images

Normally, the MCs are very small with sizes of 0.1–1.0 mm and the average size is about 0.3 mm, so some isolated ones of them are smaller than 0.1 mm, and cannot be distinguished in the film-screen mammography through high-frequency noise influence. And if to this we add the problems of visibility for tissue type of the MG, the problem becomes more complex. The radial lesions or spiculated masses have a

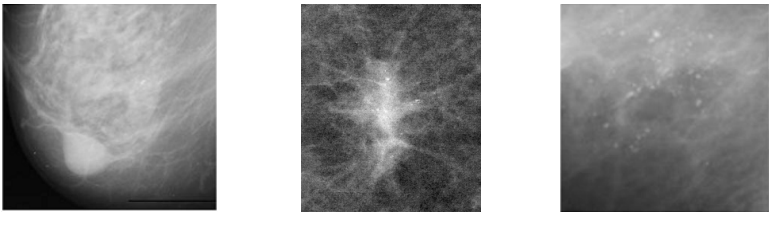


Fig. 2. Patterns: circumscribed mass (left), spiculated mass (center), and microcalcifications (right)

centered region with segments leaving it in many directions. The circumscribed masses lesions are more uniform, resembling a circle, although still irregular. The typical MG images are shown in Fig. 2.

2.3 ROIs Generation

There are many approaches for enhancement of MC clusters and masses, including various filtering methods, global and local thresholding methods, histogram equalization, mathematical morphology transformations, high-order statistic methods WT, multiresolution processing approaches, ANN, fractal models, fuzzy logic approaches, etc. After applications such the methods, the regions of interests selected from the digitized MG are de-noised and enhanced. Here, the patterns were preprocessed obtaining the negative image before applying WT with different Wavelet functions.

2.4 Wavelet Transform and Filter Banks

The Discrete Wavelet Transform (DWT) is easy to realize using filter banks [9], DWT can be implemented applying some equations, but it is usually made using filter bank techniques. The most popular scheme of the DWT for 2-D signal uses only two filters for rows and columns, as in the symmetric filter bank.

The classical wavelets applied in the experiments of this work were *Daubechies 4*, *Daubechies 8*, and *Daubechies 16*; additionally the wavelet based on *eup(x)* atomic function. The implementation of WT given by Mallat [10] is used in the proposed approach. Applying convolution of low and high pass filters on the original data, the signal can be decomposed in specific sets of coefficients, at each level of decomposition, as the component LL contains the information of low frequencies in horizontal and vertical orientations. The component HH contains the information of high frequencies in horizontal and vertical orientations. The component LH contains the information of low frequencies in horizontal orientation and high frequencies in vertical orientation. The component HL contains the information of high frequencies in horizontal orientation and low frequencies in vertical orientation.

The figures 3 and 4 show the coefficients of the decomposition low pass and high pass filters for Wavelet *Daubechies 8* and WAF *eup(x)*, respectively. These coefficients were employed in classification of the MG patterns. Procedure WT (see Fig.1)

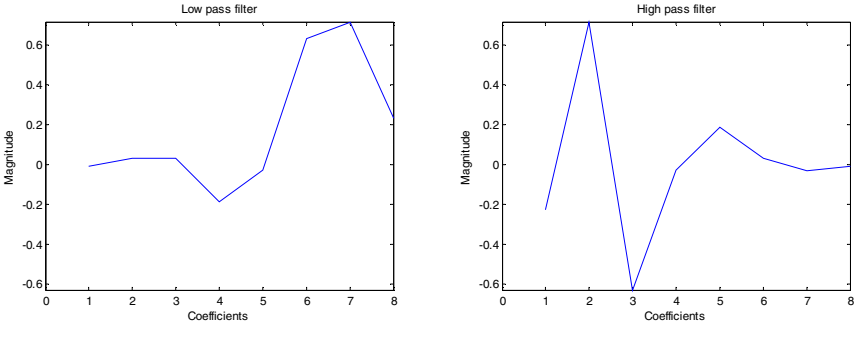


Fig. 3. Coefficients of the decomposition filters (low pass and high pass) for Wavelet *Daubechies 8*

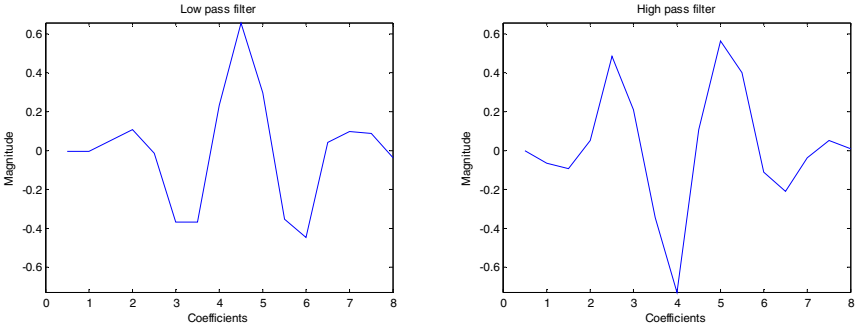


Fig. 4. Coefficients of the decomposition filters (low pass and high pass) for Wavelet based in AF *eup(x)*

is applied to enhance data before patterns detection stage permitting better distinction of the patterns from the background image.

Wavelet Key Properties. The Wavelet decomposition algorithm uses two analysis filters $\tilde{H}(z)$ (lowpass) and $\tilde{G}(z)$ (highpass).

Additionally, we present two important properties for Wavelet families applied in the classification scheme; the frequency response and projection Cosine that justify the experimental results.

Frequency response. This characteristic allows determining the behavior of the analysis and synthesis filters in a graphic way to appreciate the differences that there are among the different Wavelet families used.

Projection Cosine. The (generalized) projection angle θ *between the synthesis and analysis subspaces* V_a *and* $\tilde{V}_a$ *is defined as [11].*

$$\cos\theta = \inf_{f\in\tilde{V}_a} \frac{\left\|\mathrm{P}_a f\right\|_{L^2}}{\left\|f\right\|_{L^2}} = \frac{1}{\sup\limits_{\omega\in[0,2\pi]} \sqrt{a_{\varphi}(\omega)\cdot a_{\tilde{\varphi}}(\omega)}} \tag{1}$$

This fundamental quantity is scale-independent, and it allows to comparing the performance of the biorthogonal projection $\tilde{\mathrm{P}}_a$ with that of the optimal least squares solution P_a for a given approximation space V_a. Specifically, we have the following sharp error bound used in numerical analysis:

$$\forall f \in L^2, \left\|f-\mathrm{P}_a f\right\|_{L^2} \le \left\|f-\tilde{\mathrm{P}}_a f\right\|_{L^2} \le \frac{1}{\cos\theta}\left\|f-\mathrm{P}_a f\right\|_{L^2} \tag{2}$$

The projection angle θ between the synthesis and analysis subspaces should be 90 degrees in orthogonal spaces. In other words, the biorthogonal projector $\tilde{\mathrm{P}}_a$ could be

essentially as good as the optimal one (orthogonal projector onto the same space) when the value $\cos\theta$ is close to one [12].

3 Experimental Results

The proposed classification system has been investigated and justified on a database MIAS. The aim of the experiments was to verify the effectiveness of the scheme in classifying the clusters of MCs, SM, and CM. There were considered the extracted patterns of original image for the classification of patterns.

The figure 5 present the frequency response for Wavelet Daubechies 4, Daubechies 8, Daubechies 16 and Wavelet based in AF eup(x) showing sufficiently better frequency selection property of the last one.

Table 1 presents the key properties of the different Wavelets used in the classification scheme of the MG images

Table 1. Key properties for different Wavelet filters

Type	*Daubechies 4* Descomposition	*Daubechies 8* Descomposition	*Daubechies 16* Descomposition	AFW *eup(x)* Descomposition
Approximation Order	2	4	8	4
Projection Cosine	0.979	0.988	0.983	0.997

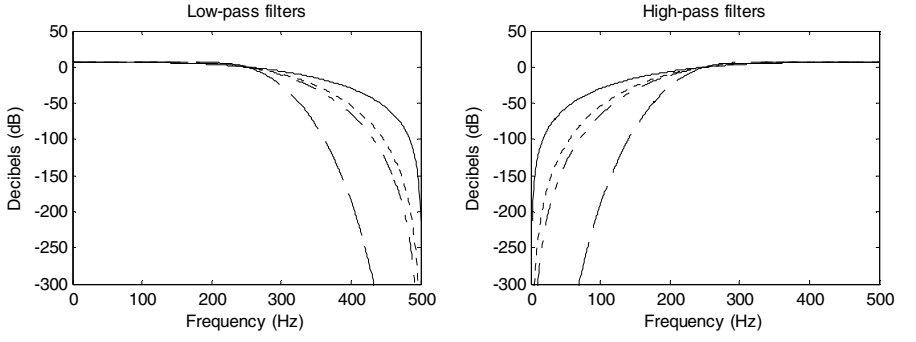


Fig. 5. Frequency response in decibels of the decomposition low-pass and high-pass filters, continuous line is for Wavelet *Daubechies 4*, doted line is for Wavelet *Daubechies 8*, dashdot line is for Wavelet *Daubechies 16*, and dashed line is for Wavelet based in AF *eup(x)*

It is known that the approximation property of estimation of random variable is characterized by relative error $\delta = 2(1-r)$, where r is correlation coefficient that is equal to projection cosine in this case. So, calculations of this error show that Wavelet based on *eup(x)* can potentially produce relative variance error of 0.00464 (6.8% in RMS value), and in same time Wavelet *Daubechies 8* gives value of 0.02242 (more than 15% in RMS value). So, Wavelet based on *eup(x)* realizes sufficiently better

approximation than Wavelet *Daubechies 8* that presents the better result for classical Wavelet families.

The results exposed in Fig. 6-9 show the experimental results applying different number of nodes in hidden layer of classifier. One can see that in the case of classification between normal, microcalcifications, spiculated masses, and circumscribed

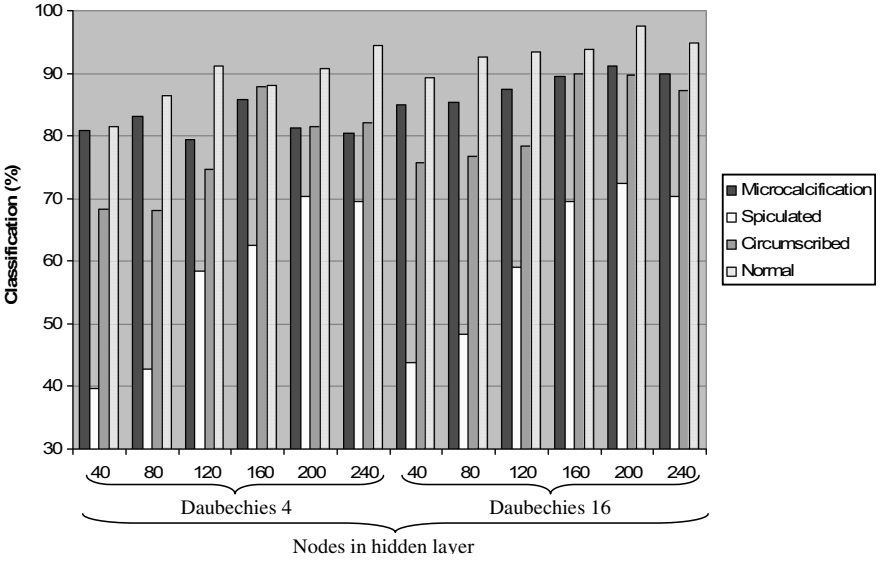


Fig. 6. Classification Percentage on Pattern Type for *Daubechies 4* and *Daubechies 16*

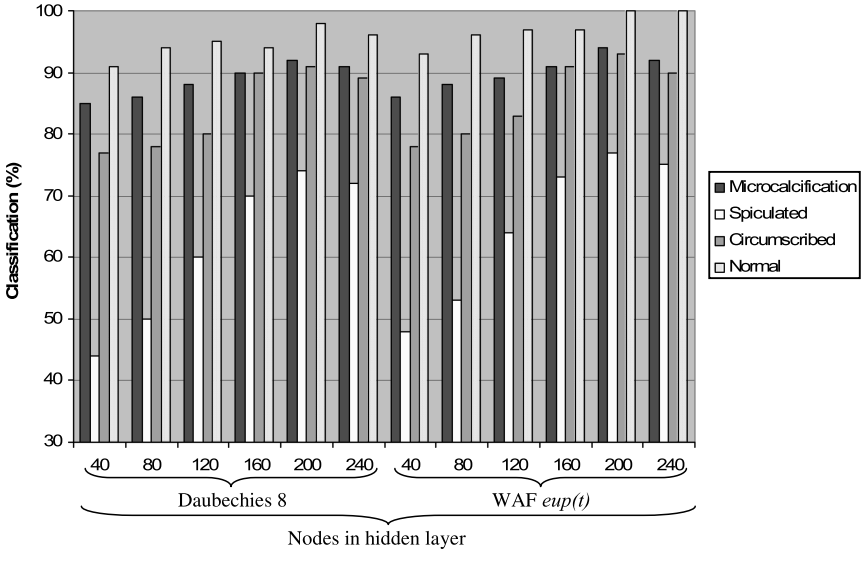


Fig. 7. Classification Percentage on Pattern Type for *Daubechies 8* and WAF

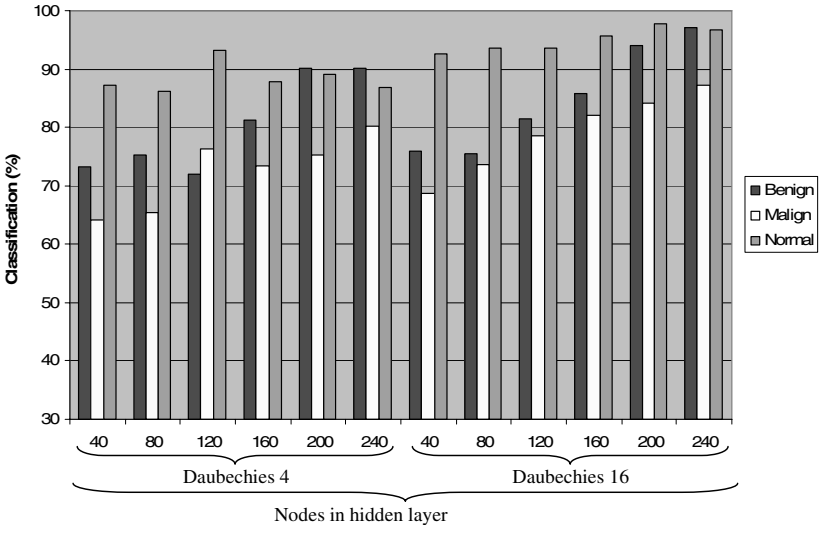


Fig. 8. Classification Percentage on Tumor Type for *Daubechies 4* and *Daubechies 16*

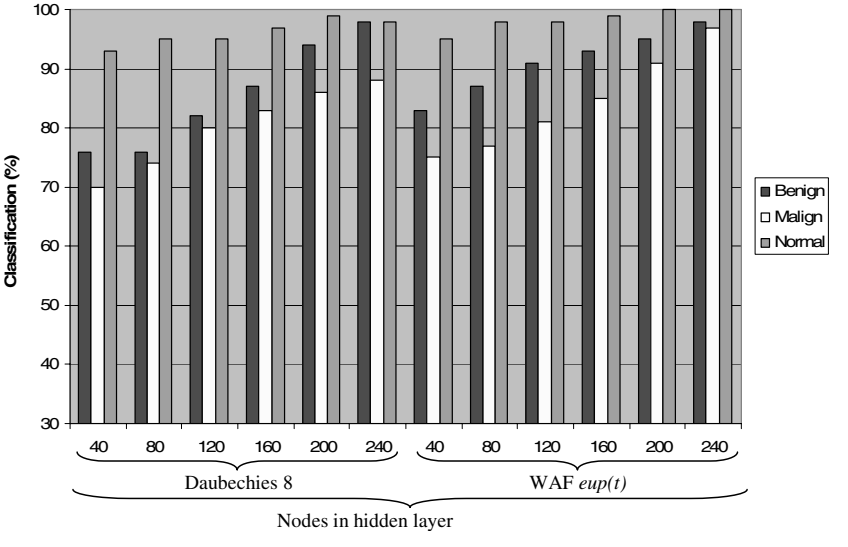


Fig. 9. Classification Percentage on Tumor Type for *Daubechies 8* and WAF

masses areas, the classification levels are similar for different wavelet basis, *Daubechies 8* and WAF. All of the MGs belonging to MC and normal classes were correctly classified; the high correct classification percentages were 94% and 100%, respectively. For spiculated and circumscribed masses, the high correct classification percents were 77% and 93%, respectively. When the number of nodes used in hidden layer of the classifier increases the successful classification rates also overall enlarge.

Fig. 9 exposes the successful rates concerning the results obtained for tumor type, for the benign class the successful rate achieved was 98% for both Wavelets used. For the malign class, the successful rate achieved was 88% using *Daubechies 8* and 97% using *eup(x)*. One can see that 93% and 100% of the patterns in normal class were correctly classified using *Daubechies 8* and *eup(x)*, respectively.

Finally, the numerous tests of classification rates have shown that Wavelet based on the atomic function *eup(x)* presents better recognition properties (see Fig. 7 and Fig. 9) in comparison with other wavelets used. This result can be justified analyzing Fig. 5, where more selective frequency response for *eup(x)* can be observed. Also, the projection cosine for last WAF is near to optimal (Table 1), this implies that it is "better semi-orthogonal" in comparison with other wavelets, which permits to approximate data better.

4 Conclusions

The proposed scheme is based on Wavelet Transform and MLP classifiers for MG medical image. The experimental results have shown sufficiently good performance of the implemented algorithms. We present a way to construct a solution considering the two main classification problems of mammograms (i.e. nature of tumor and related geometries). It has been also evaluated the proposed method on real data showing promising rate of recognition benign, malign and normal MGs. The implemented scheme permits to reduce the iterations number during the training of the ANN MLP applying WT. We adapted Daubechies (*Daubechies 4*, *Daubechies 8*, and *Daubechies 16*) and Wavelet based on atomic function (*eup(x)*) using MLP network for MCs, spiculated and circumscribed masses classification in the MG images. In the future work, other wavelets based on atomic functions type can be applied in the proposed scheme to achieve better classification results.

Acknowledgments. The authors thank the National Polytechnic Institute, CONACYT, and UAEM-PROMEP of Mexico by its support.

References

1. Jan, J.: Medical Image Processing, Reconstruction and Restoration: Concepts and Methods. Taylor & Francis, USA (2006)
2. Papadopoulos, A.N., Plissiti, M.E., Fotiadis, D.I.: Medical-Image Processing and Analysis for CAD Systems. In: Costaridou, L. (ed.), pp. 51–79. Taylor & Francis/CRC Press (2005)
3. Ruiz, V.F., Nasuto, S.J.: Biomedical-Image Classification Methods and Techniques. In: Costaridou, L. (ed.), pp. 137–176. Taylor & Francis/CRC Press (2005)
4. Costaridou, L.: Medical image analysis methods. CRC Press, USA (2005)
5. Rashed, E.A., Ismail, I.A., Zaki, S.I.: Multiresolution mammogram analysis in multilevel decomposition. Elsevier Pattern Recognition Letters 28, 286–292 (2007)
6. Juarez, C., Ponomaryov, V.: Classification of Microcalcifications using Wavelet Transform and neural Networks. GESTS Intern. Trans. on Comp. Sciens. and Eng. 39(1), 112–120 (2007)

7. Gulyaev, Y.V., Kravchenko, V.F., Pustovoit, V.I.: A New Class of WA-Systems of Kravchenko–Rvachev Functions. Doklady Mathematics 75(2), 325–332 (2007)
8. Mammographic Image Analysis Society (MIAS), MiniMammography Database, http://www.wiau.man.Ac.uk/services/MIAS/MIASmini.html
9. Jähne, B.: Practical Handbook on Image Processing for Scientific and Technical Applications. CRC Press, USA (2004)
10. Mallat, S.G.: A theory for multiresolution signal decomposition: The wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11, 357–381 (1989)
11. Unser, M., Aldroubi, A.A.: General sampling theory for nonideal acquisition devices. IEEE Trans. Signal Process. 42(11), 2915–2925 (1994)
12. Strang, G., Nguyen, T.Q.: Wavelets and Filter Banks. Wellesley-Cambridge Press, Cambridge (1996)

The Effect of Repeated Measurements on Bayesian Decision Regions for Class Discrimination of Time-Dependent Biological Systems

Arturo Baltazar[1] and Jorge I. Aranda-Sanchez[2]

[1] Centro de Investigación y Estudios Avanzados,
CINVESTAV Unidad-Saltillo,
Robotics and Advanced Manufacturing Program,
Carretera Saltillo-Monterrey Km 13.5,
Ramos Arizpe 25900, Coahuila, México
arturo.baltazar@cinvestav.edu.mx
[2] Facultad de Ciencias Físico-Matemáticas
Universidad Michoacana de San Nicolás de Hidalgo (UMSNH)
Morelia 58070, Michoacán, México
jorge_isidro@yahoo.com

Abstract. Nowadays, it is common to use nondestructive sensors to monitor property variations in biological systems. The repeated observations on the time varying system are referred to as repeated measurements. In many applications, it is important to develop a Bayesian classifier based on repeated measurements data to assure proper class identification. However, its implementation is complex due to the multidimensional and discontinuous nature of the decision boundaries. In this work, the problem of correlated data to develop a Bayesian Classifier for a multiclass problem is addressed. The effect of correlation on the classification error rate is discussed. It was found that additional correlated data does not improve the classifier likelihood for highly correlated repeated measures. Also, it is shown that error classification is adversely affected by correlation between repeated measures. Finally, a strategy to develop a multiclass Bayesian classifier from multisensory repeated measurements data is presented.

Keywords: Bayesian classifier; repeated measurements; nondestructive sensors; multiclass problem.

1 Introduction

The Bayesian strategies for pattern classification minimize the expected risk [1]. The probability functions for each class define decision regions in which boundaries with the highest probability of misclassification can occur. According to the Bayes decision rule to assign a vector pattern x to a class w_i, with minimum error can be expressed as

$$p(w_i)\,p(x|w_i) > p(w_n)\,p(x|w_n) \quad n,i = 1,...,a; n \neq i, \tag{1}$$

where $p(w_1),\ldots, p(w_n)$,are known prior probabilities.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 305–314, 2008.

In this approach, to design an optimal classifier, it is assumed that the forms of the underlying density functions are known. At the same time, the parametric Bayesian approach for classification requires the correct definition of the probability density functions for each class.

Integration of sensor data using data fusion can be either complementary or used to enhance the response of a single sensor. The use of data fusion in robotics spans from improving position location and distance assessment to pattern recognition combining information from different sensors [2]. Attempts to solve the problem of classification and modeling using multisensors on time dependent systems have not been completely successful. Finding parametric distribution functions is rather complex when sensor data fusion is considered.

Additionally, the nondestructive data taken from a group of samples over time results in a collection of correlated data usually referred to as repeated measurements data. The statistical analysis has been thoroughly studied recently in different research areas that deal with biological systems ([3], [4], [5], [6]). In most of these studies the focus is more on the statistical analysis of treatment effect between groups, and less on the description of the probability distribution as function of the time for class discrimination.

The aim of this work is threefold, first to show how sensor data fusion reduces the probability of error; second to develop a scheme for a Bayesian classifier using non independent data from repetitive nondestructive measurements on time dependent biological systems; and third to study the effect of correlation and the number of repetitive nondestructive tests on the classification error.

2 Bayesian Classifier Using Repeated Measures

In most cases it can be correctly assumed that the experimental data series needed for the probability density definition arose from a stochastic process. In the particular case where x represents a vector of repeated measurements, this assumption is violated. Data from repeated measurement represents observations per object (experimental unit), collected at specific times, that gives information about the development of a response (characteristic) Y. Therefore, the response for a specific object is correlated in time even though the measurements obtained from different subjects are independent.

In general, for a response vector Y_{kj} with normal distribution, its symmetric variance-covariance matrix can be given as [7]

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1p} \\ \sigma_{12} & \sigma_{22} & \dots & \sigma_{2p} \\ \vdots & \vdots & & \vdots \\ \sigma_{1p} & \sigma_{2p} & \dots & \sigma_{pp} \end{bmatrix}, \tag{2}$$

where σ_{kj} is the covariance between different objects and measurements.

To simplify the problem it is reasonable to assume a matrix with compound symmetry pattern ([7], [8]). The resulting covariance matrix for every object is given as

$$\Sigma_p = \sigma^2 \left[(1-\rho) I_p + \rho J_p \right], \tag{3}$$

where I_p is a p x p identity matrix, $J_p = 1_p 1_p^T$ is a p x p matrix with unit components.

Assuming that the underlying probability density function is multinomial, using the covariance matrix for one measurement parameter, x, this function is given as

$$p(x|w_i) = \frac{1}{(2\pi)^{3/2} \sqrt{|\Sigma|^2}} \exp\left(-\frac{1}{2} (x-\mu_i)^T \Sigma^{-1} (x-\mu_i) \right), \tag{4}$$

where w_i is a specific class, and $|\Sigma|$ is the determinant of the covariance matrix Σ and Σ^{-1} its inverse, $\mu_i = [\mu_{i1}, ..., \mu_{in}]$, where each component of vector μ_i corresponds to the mean value at a specific repeated measurement n; μ_i^T is the transpose of the μ_i vector; and the input x vector with equal vector components ($\bar{x}$) is defined as $x = \bar{x} 1_p$.

In this context, multisensor data fusion approach in a multiclass problem can therefore be defined as

$$p(y|w_n) = \prod_{s=1}^{r} p_s(x_s|w_n), \; n = 1, 2, 3 \ldots a, \tag{5}$$

where w_i is the class, a is the number of classes, r is the number of characteristics, $y = [x_1, x_2, ..., x_r]$ is a pattern vector which corresponds to the fused data, x_s ($s = 1, \ldots, r$) is a one-dimensional variable and $p_s(x_s|w_n)$ is the probability density function for each characteristic or in our case a measurement parameter (Eq.(5)) [9].

The probability of error for a classes is given as $p(error|y) = \min\left[p(w_1|y), p(w_2|y), \cdots, p(w_a|y) \right]$ and the average probability of error is given in terms of decision regions in which the vector x falls [1]

$$p(error) = \sum_{n=1}^{a} \int_{R_n} p(w_n)\, p(y|w_n)\, dx, \tag{6}$$

which is known as Bayes risk [10] or probability of error, where $y \in R_n$ and satisfies the following equation $R_n = \{ y \mid p(w_i) p(y|w_i) < \min_{\substack{1 \le i \le n \\ i \ne n}} \{ p(w_n) p(y|w_n) \} \}, i \ne n$.

The exact calculation of Eq. (6) for a multiclass and multivariate problem is quite difficult, even if normal distribution is assumed. This is due to the discontinuous nature of the decision regions. An approach, based on Bhathacharyya's bound is discussed in reference [11] for a two class problem and it is shown in Fig. 1. It was found that increasing the number of sensors (multivariate case) considerably reduces the probability of error.

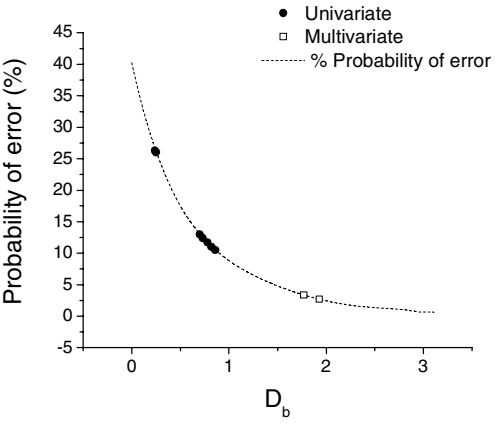


Fig. 1. Probability of error in terms of the Bhattacharyya distance. The estimate of the probability error using experimental data is shown by the filled circles for the univariate case and by open squares for the multivariate case. The theoretical error estimation is shown by the dashed line [11].

3 Effect of Correlation on the Bayesian's Decision Regions

A discriminant function $g(y)$ defines a decision region assigning the input vector y to a class *i* if

$$g_i(y) > g_n(y) \quad \forall i \neq n,\ i,n = 1,2,...,a\,. \tag{7}$$

Then, Bayes' rule can be implemented in terms of discriminant functions as

$$g_i(y) = \frac{p(y|w_i)\,p(w_i)}{\sum_{n=1}^{a} p(y|w_n)\,p(w_n)}\,. \tag{8}$$

Each discriminant function generates a decision regions. These decision regions, R_n, are separated by decision boundaries which satisfy Bayes rule (Eq. (8)).

The decision boundary satisfies $g_i(y) = g_n(y)$, and $y \in R_i$ if Eq. (8) holds. The discriminant function (Eq. (8)) can be written in the following form [1]

$$g_i(y) = y^T W_i\, y + U_i^T\, y + u_{i0}\,, \tag{9}$$

where

$$\begin{aligned} W_i &= -\frac{1}{2}\Sigma_i^{-1}, \\ U_i &= \Sigma_i^{-1}\mu_i\,, \\ u_{i0} &= -\frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i - \frac{1}{2}\ln\left(|\Sigma_i|\right) + \ln\left(p(w_i)\right). \end{aligned} \tag{10}$$

Eq. (9) represents the general case for a multiclass, multivariate case, where the covariance Σ_i can have any value. The presence of correlation within the covariance

matrix makes defining the decision region even more complicated. However, repeated measurements can be studied using simple patterns of covariance matrices such as compound symmetry (Eq. (3)) [12]. Before the problem of repeated measurements is addressed, it is helpful to study the case of two multinomial independent classes with the discriminant function satisfying

$$g_1(y) = g_2(y)\,, \tag{11}$$

where

$$y = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},\ \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix},$$

$$\mu' = \begin{pmatrix} \mu_3 \\ \mu_4 \end{pmatrix},\ \Sigma_1 = dig(\sigma_1^2, \sigma_2^2)\,,\ \Sigma_2 = diag(\sigma_3^2, \sigma_4^2)\,, \tag{12}$$

$$\Sigma_1^1 = diag(\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2})\,, \Sigma_2^1 = diag(\frac{1}{\sigma_3^2}, \frac{1}{\sigma_4^3})\,.$$

Substituting in Eq. (11),

$$g_1(y) = y^T W_1\, y + \omega_1^T\, y + u_{10}\,,$$
$$g_2(y) = y^T W_2\, y + \omega_2^T\, y + u_{20}\,,$$

where

$$W_1 = -\frac{1}{2}\Sigma_1^{-1},\ \omega_1 = \Sigma_1^{-1}\mu\,,$$
$$u_{10} = -\frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu - \frac{1}{2}\ln\left(|\Sigma_1|\right) + \ln\left(P(w_1)\right), \text{ and} \tag{13}$$
$$W_2 = -\frac{1}{2}\Sigma_2^{-1},\ \omega_2 = \Sigma_2^{-1}\mu'\,,$$
$$u_{20} = -\frac{1}{2}\mu_2^T\Sigma_2^{-1}\mu' - \frac{1}{2}\ln\left(|\Sigma_2|\right) + \ln\left(P(w_2)\right).$$

When $\Sigma_1 \neq \Sigma_2$, and after some algebra, Eq. (11) becomes a quadratic function as follows

$$A\,x_1^2 + C\,x_2^2 - 2B\,x_1 - 2D\,x_2 + E + F + 2G = 0\,, \tag{14}$$

where

$$A = \frac{1}{\sigma_1^2} - \frac{1}{\sigma_3^2},\ C = \frac{1}{\sigma_2^2} - \frac{1}{\sigma_4^2},\ B = \frac{\mu_1}{\sigma_1^2} - \frac{\mu_3}{\sigma_3^2},$$
$$D = \frac{\mu_2}{\sigma_2^2} - \frac{\mu_4}{\sigma_4^2},\ E = 2\frac{\mu_1^2}{\sigma_1^2} - 2\frac{\mu_3^2}{\sigma_3^2},\ F = 2\frac{\mu_2^2}{\sigma_2^2} - 2\frac{\mu_4^2}{\sigma_4^2},\ G = \ln\left(\frac{\sigma_1\sigma_2}{\sigma_3\sigma_4}\right). \tag{15}$$

Thus, Eq. (14) is a quadratic equation with two variables representing a conic section. A parameter $\Delta = -4AC$ can be defined such as if $\Delta < 0$ the equation represents an ellipse, $\Delta = 0$ the equation represents a parabola and for $\Delta > 0$ a hyperbola, finally when $A = C$ a circle.

If a covariance matrix with compound symmetry pattern is assumed with the form

$$\Sigma_{1\rho} = \sigma_1^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix},\ \Sigma_{2\rho} = \sigma_2^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix},$$
$$\Sigma_{1\rho}^{-1} = \frac{1}{\sigma_1^2 (1\text{-}\rho)} \begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix},\ \Sigma_{2\rho}^{-1} = \frac{1}{\sigma_2^2 (1\text{-}\rho)} \begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix}. \tag{16}$$

Then, for the case when $\Sigma_{1\rho} = \Sigma_{2\rho}$ the corresponding linear discriminant function is given by

$$\frac{2}{\sigma_1^2 (1+\rho)}[(\mu_2 - \mu_1)\,\bar{y} - (\mu_2^2 - \mu_1^2)] = 0\,, \tag{17}$$

where $\bar{y}$ is the average of the input vector components.

If $\Sigma_{1\rho} \neq \Sigma_{2\rho}$, then, using Eq. (11), Eq. (12) and Eq. (13) can now be expressed as,

$$A_1 x_1^2 + B_1 x_1 x_2 + C_1 x_2^2 + D_1 x_1 + E_1 x_2 + F_1 = 0\,, \tag{18}$$

where

$$A_1 = \frac{1}{2(1\text{-}\rho^2)}\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right),\ B_1 = \frac{\rho}{1\text{-}\rho^2}\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}\right),$$
$$C_1 = \frac{1}{2(1\text{-}\rho^2)}\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right),\ D_1 = \frac{1}{1\text{-}\rho^2}\left(-\frac{\mu_1}{\sigma_1^2} + \frac{\mu_2 \rho}{\sigma_1^2} + \frac{\mu_3}{\sigma_2^2} - \frac{\mu_4 \rho}{\sigma_2^2}\right),$$
$$E_1 = \frac{1}{1\text{-}\rho^2}\left(-\frac{\mu_2}{\sigma_1^2} + \frac{\mu_1 \rho}{\sigma_1^2} + \frac{\mu_4}{\sigma_2^2} - \frac{\mu_3 \rho}{\sigma_2^2}\right), \text{ and} \tag{19}$$
$$F_1 = 2\ln\left(\frac{\sigma_1}{\sigma_2}\right) + \frac{1}{1\text{-}\rho^2}\left(\frac{\mu_1^2}{\sigma_1^2} + \frac{\mu_2^2}{\sigma_1^2} - \frac{2\mu_1\mu_2\rho}{\sigma_1^2} - \frac{\mu_3^2}{\sigma_2^2} - \frac{\mu_4^2}{\sigma_2^2} + \frac{2\mu_3\mu_4\rho}{\sigma_2^2}\right).$$

In this case, a parameter $\Delta_1 = B_1^2 - 4A_1C_1$ can be defined in Eq. (18); as above, if $\Delta_1 < 0$ the equation corresponds to an ellipse, if $\Delta_1 = 0$ a parabola and if $\Delta_1 > 0$ a hyperbola.

In general, after some algebra $\Delta_1 = -\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right)^2 \frac{1}{1\text{-}\rho^2}$ represents an ellipse; this result is valid for all values of ρ in the interval $[0,1)$.

To show how the global correlation affects the form of the discrimination functions and classification error, in the case of two classes, a compound symmetry model is used. Considering the case of *n* repeated measurements, the covariance matrices are defined as $\Sigma_{1\rho} = \sigma_1^2 \Gamma$ and $\Sigma_{2\rho} = \sigma_2^2 \Gamma$, where Γ has unit diagonal elements and off-diagonals ρ. The corresponding results for inverse matrices are $\Sigma_{1\rho}^{-1} = \Gamma^{-1}/\sigma_1^2$ and $\Sigma_{2\rho}^{-1} = \Gamma^{-1}/\sigma_2^2$, where $\Gamma^{-1} = \{1/[\sigma_1^2(1\text{-}\rho)]\}\,\{I\text{-}JJ^T\rho/[1+(n\text{-}1)\rho]\}$, with *J* as a vector with *n* unit components. Using Eq. (11) and Eq. (13) with $y = (x_1,...,x_n)$, $\mu = \mu_1 J$, $\mu = \mu_2 J$, where *J* is a vector with *n* unit components, the corresponding discriminant function is

$$[\frac{1}{2\sigma_2^2(1-\rho)}-\frac{1}{2\sigma_1^2(1-\rho)}][\|y\|^2-\frac{n\bar{y}\rho}{1+(n-1)\rho}]+\frac{1}{2}Ln[\frac{|\Sigma_{1\rho}|}{|\Sigma_{2\rho}|}]+$$
$$\frac{1}{(1-\rho)}[(\frac{\mu_2}{\sigma_2^2}-\frac{\mu_1}{\sigma_1^2})\bar{y}-\frac{1}{2}(\frac{\mu_2^2}{\sigma_2^2}-\frac{\mu_1^2}{\sigma_1^2})][n-\frac{\rho}{1+(n-1)\rho}n^2]=0, \quad (20)$$

where $|\Sigma_{1\rho}|=\sigma_1^{2n}f(\rho)$ and $|\Sigma_{2\rho}|=\sigma_2^{2n}f(\rho)$ are the determinants of covariance matrices and $f(\rho)$ is a polynomial with maximum degree *n*; $\|y\|$ is the norm of vector y with *n* components. In this case it is difficult to define a parameter like Δ_1 because Eq. (20) is multifunctional. If the covariances of both classes are equal, then, only the second term in Eq. (20) remains,

$$\frac{1}{(1-\rho)}[(\frac{\mu_2}{\sigma_2^2}-\frac{\mu_1}{\sigma_1^2})\bar{y}-\frac{1}{2}(\frac{\mu_2^2}{\sigma_2^2}-\frac{\mu_1^2}{\sigma_1^2})][n-\frac{\rho}{1+(n-1)\rho}n^2]=0\ . \quad (21)$$

4 Numerical Estimations

In Fig. 2, the decision region (Eq. (21)) is described as function of correlation factor and number of repeated measurements (*n*). It is shown that for strongly correlated data, the number of repeated measurements does not have an influence on the decision regions. In other words, no new information is added to the classifier and few repeated measurements are needed to establish the Bayesian classifier.

The boundary decision (Eq. (20)) describes a circle when $\rho=0$ and an ellipse when $\rho\neq 0$ .To study the effect when covariance are not equal, data from [11] was used. The experimental data belongs to two different and independent sensors, a

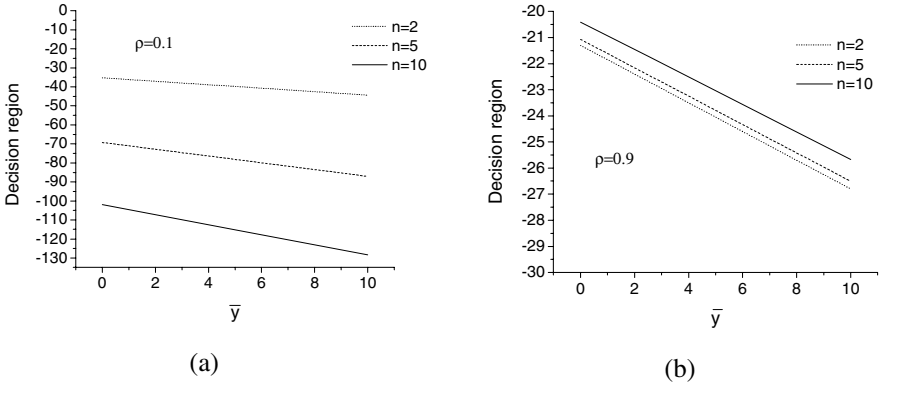


Fig. 2. Effect of correlation factor on ρ and *n* on the decision region as given in Eq. (21); (a) ρ=0.1 and (b) ρ=0.9

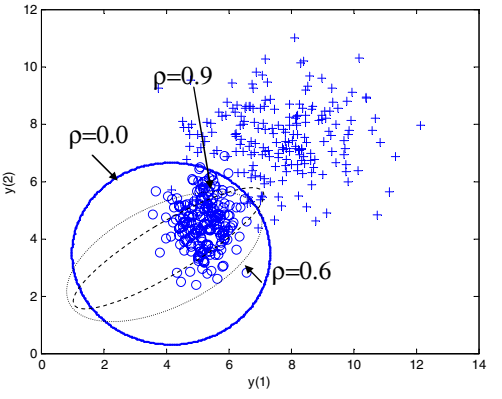


Fig. 3. Effect of correlation factor (ρ) on the decision region of a Bayesian multivariate classifier with compound symmetry pattern as described in Eq. (20) with n=2; Scatter points are synthetic data, (o) Class 1, (+) Class 2

sound and a colorimeter sensor which provided measurements of acoustic stiffness(Sc) and color, respectively. Measurements were used to monitor ripening in tomato fruit samples at specific times (repeated measurements). The scatter points in Fig. 3 are synthetic data for Sc obtained from reconstructed multinomial probability distributions using real experimental values of mean and variance of repeated measurements data taken from two testing days to indicate one of the two classes.

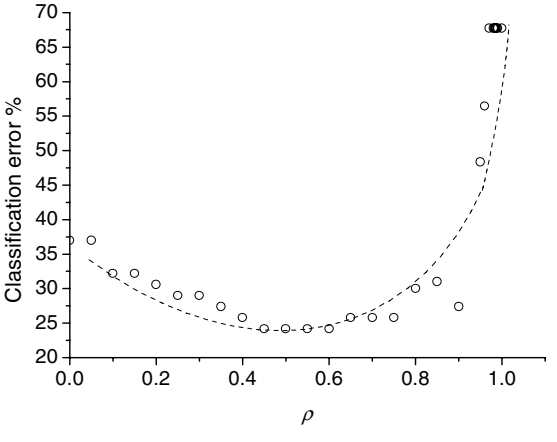


Fig. 4. Effect of correlation factor (ρ) on the probability of error for a multivariate classifier assuming compound symmetry and n=3 number of repeated measurements

Results show that the classification error is being affected by the correlation by incorrectly classifying the scatter points that belong to class 1 (inside ellipse) as belonging to class 2. The classification error increases monotonically with correlation.

To investigate how correlation factor affects the classification error, Eq. (5) with a covariance matrix having compound symmetry pattern (Eq. (20)) was used. Data from *Sc* and color [11] gathered from three repeated measurements (n=3) were used. After building the classifier with training data, new testing data was input and results obtained (Fig. 4). It is shown that, if data from repeated measurements is incorrectly assumed as independent (ρ=0) the classification error is increased. There is, however, a minimum probability of error when ρ is about 0.6 near the value of the average correlation factor found in the experimental data used for training the classifier.

5 Conclusions

This research discusses the effect of data correlation typically found in repeated measurements on the development of a Bayesian classifier. A scheme for data fusion used to describe the parametric multivariate probability density function required by a Bayesian classifier was discussed. It was found that the probability of error is reduced largely when the number of sensor (fused data) increases. Decision regions were found to be affected by the number of repeated measurements especially in the presence of highly correlated data. The importance of considering a covariance matrix, for correlated data, was demonstrated theoretical and numerically using actual data reported in the literature. It was shown that if the training data is incorrectly assumed as independent,t the classification error is larger than when a correlation factor is considered.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, Chichester (2001)
2. Kokar, M., Tomasik, J.A., Mieczyslaw Weyman, J.: Formalizing classes of information fusion systems. Information Fusion 5(4), 189–202 (2004)
3. Verbeke, G.: Linear Mixed Models for Longitudinal Data. Springer, New York (2000)
4. Lindsey, J.K.: Models for Repeated Measurements. Oxford Statistical Science Series. Oxford University Press, Oxford (2005)
5. Lindsey, J.K., Lindsey, P.J.: Multivariate distributions with correlation matrices for nonlinear repeated measurements. Computational Statistical & Data Analysis 50, 720–732 (2006)
6. Molenberghs, G., Verbeke, G.: Meaningful statistical model formulations for repeated measures. Statistica Sinica 14, 989–1020 (2004)
7. Hearne, E.M., Clark III, G.M., Hatch, J.P.: A test for serial correlation in univariate repeated-measures analysis. Biometrics 39, 237–243 (1983)
8. Toutenburg, H.: Statistical analysis of designed experiments, 2nd edn. Springer, Heidelberg (2002)
9. Krzysztofowicz, R., Long, D.: Fusion of detection probabilities and comparison of multisensor systems. IEEE Transactions on Systems, Man and Cybernetics 20(3), 665–670 (1990)

10. Garber, F.D., Djouadi, A.: Bounds on the Bayes classification error based on pairwise risk functions. IEEE Trans. pattern analysis and machine intelligence 10(2), 281–288 (1988)
11. Baltazar, A., Aranda, J.I., Gonzalez-Aguilar, G.: Bayesian classification of ripening stages of tomato fruit using acoustic impact and colorimeter sensor data. Computers and Electronics in Agriculture 60(2), 113–121 (2008)
12. Andrews, D.F., Brant, R., Percy, M.E.: Bayesian incorporation of repeated measurements in logistic discrimination. The Canadian Journal of Statistics 14(3), 263–266 (1986)

Global Classifier for Confidential Data in Distributed Datasets

Omar Jasso-Luna, Victor Sosa-Sosa, and Ivan Lopez-Arevalo

Center for Research and Advanced Studies
Laboratory of Information Technology
Cd. Victoria, Tam. Mexico
{jjasso,vjsosa,ilopez}@tamps.cinvestav.mx

Abstract. Everyday, a huge amount of data are produced by many institutions. In most of the cases these data are stored on centralized servers where usually are analyzed to extract knowledge from them. This knowledge is represented by patterns or tendencies that become valuable assets for decision makers. Data analysis requires high performance computing. This situation has motivated the development of Distributed Data Mining (DDM) architectures. DDM uses different distributed data sources to build a global classifier. Building a global classifier implies that all of the data sources be integrated in a unique global dataset. This means that private data have to be shared by every participant. This situation sometimes represents a data privacy intrusion that is not desired by data owners. This paper describes a DDM application where participants work in an interactive way to built a global classifier for data mining process without need sharing the original data. Results show that the global classifier created of this way offers better performance than doing it individually and avoids data privacy intrusion.

Keywords: Data Mining, Classification.

1 Introduction

Nowadays complex business and industrial-scientific applications require storing a huge amount of data everyday. In most of the cases this huge organizations tend to build centralized dataset or Data warehouse that collects all information from their geographically dispersed branches. This data is analysed to discover patterns or tendencies that represent knowledge, which is an important asset for any organization. Many organizations carry out the analysis of data employing Data Mining (DM) techniques. There are some toolkits that implement these techniques. One of them is Weka[1], which is a widely used DM toolkit that contains a large collection of state-of-the-art machine learning algorithms written in Java. However, mining huge centralized datasets in a stand-alone approach requires powerful equipment working with high computer resources. Distributed Data Mining (DDM) is considered a feasible strategy to help with this issue. Some architectures are exposed in [3,5,4,2]. Most of them focus in the execution of DDM tasks over remote nodes.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 315–324, 2008.

However they do not consider building a classifier without having a global knowledge of the original data. This represents an important issue when some organizations are not able to share their information. As an example, we can find some health care institutions, which want to detect some potential patients who could get a type of cancer based on his/her history data. Some of these institutions do not have enough information to build a reliable classifier which could help with this process. Some other institutions have enough information for doing this work but they are not able to share it because of their privacy policies. This paper presents an architecture which allows this type of institutions to work together for building better classifiers using global knowledge without data privacy intrusion. This architecture is based on Web technologies and Java components.

This paper is organized as follows: Section 2 describes some of related work. Section 3 shows the architecture of our approach. Section 4 describes briefly the Distributed ID3 Classifier process. Section 5 presents the implementation. Section 6 shows the preliminary results and finally, section 7 concludes the paper with ongoing and future work.

2 Related Work

DM can be defined as an infrastructure that uses a selection of different algorithms and statistical methods to find interesting and novel patterns within large datasets. It can be classified in three stages[4]. The first generation of tools provide users with a single DM algorithm operating on data stored in a local file. Examples include the use of classification algorithms such as C4.5[7], clustering algorithms such as K-means[8]. Such tools were provided primarily as standalone executable, obtaining input from the command line or via a configuration file. The second generation of tools combined a collection of different algorithms for DM under a common framework, and enabled users to provide input from various data sources. Some of these tools are described below:

- Weka[1] contains tools for classification, regression, clustering, association rules, visualization, and pre-processing.
- Illimine[10] is another DM tool developed in C language. It built-ins algorithms in data cubing, association mining, sequential pattern mining, graph pattern mining, and classification.
- Rattle[12] is a DM tool based on statistical language R[11].
- Rapid Miner[13,9] features more than 400 operators for DM in Java which can be used merging some of them.

Subsequently, third generation tools started to address the limitations that are imposed by the closed world model. Some examples of third generation tools are:

- Grid Weka[3], essentially a modification to the Weka toolkit that enables the use of multiple computational resources when performing data analysis.

- WekaG[5] is an adaptation of the Weka toolkit to a Grid environment. It is based on a client/server architecture. The server side defines a set of Grid Services that implements the functionalities of the different algorithms and phases of the DM process.
- FAEHIM[4] consists of a set of DM Web services for DDM, a set of tools to interact with this services, and a workflow system used to assemble these services and tools.
- Weka4WS[2] extends Weka to support remote execution of the DM algorithms. In such way, DDM tasks can be executed on decentralized Grid nodes by exploiting data distribution and improving application performance.

Although most of these toolkits support DDM, they focus on executing DDM tasks over remote nodes to get a benefit from all their distributed computing resources. For carrying out this process, they need to upload complete data in the processing nodes without paying attention in data privacy. Our approach is based on an architecture, which build a Global Classifier through distributed datasets. This approach pays special attention in data privacy since our Central Computing Node are not able to access original data. One potential use of this approach could be to help health care institutions to share information, getting a better classifier which allows detecting dangerous illness in a patient at early stages.

3 Architecture for Building a Global Classifier

This architecture attempts to build a Global Classifier through a distributed datasets with similar structures. This approach is based on a set of Java components executed on remotes machines, which interchange metadata with a Central Computing Node. This approach avoids the data privacy intrusion from third-party institutions by means of just interchanging metadata. The approach allows small and large institutions exchange data between them for building a better classifier that takes into account global knowledge. This architecture includes the following: a Central Computing Node (CCN) and a set of Local Nodes.

- Central Computing Node is the main component in this architecture and contains a Classifier Builder and a Global Classifiers Repository
  - A Classifier Builder is the responsible for receiving all metadata from the distributed Java components, grouping them and built the Global Classifier
  - Global Classifiers Repository contains the built classifiers
- Local Node represents a server that contains the following components:
  - Java component that accesses a private dataset and send only metadata to the CCN
  - The local dataset represents all the files like data bases or text files where the data is stored

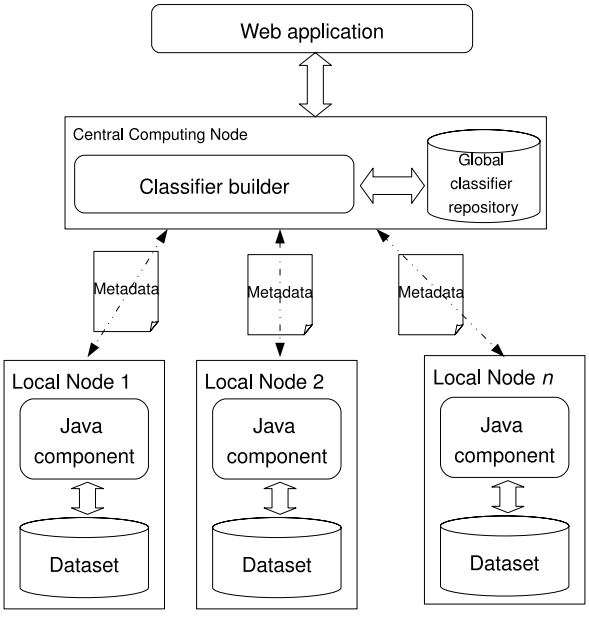


Fig. 1. Architecture

A prototype of Web application called Team Miner implements this architecture. It allows users to be registered for collaborating in a Global Classifier building process. This process consists in defining working groups where the interested institutions work together to build a global classifier. It is carried out by means of interchanging metadata between Java nodes previously defined as a team member in our CCN. CCN gathers all metadata from Java components without need to know the real data. This way to build a global classifier avoids original data intrusion and motivates organizations to participate in building a better global classifier.

3.1 Building a Global Classifier

The process of building a global classifier begins when a new local, node registered at the CCN, wants to obtain a global classifier. A registered local node asks to CCN to obtain a global classifier using the local node dataset and other datasets with similar structures located in all of registered local nodes.This process follows the next steps:

1. A Java component is registered to join to the group that study or analyze datasets with similar structures. These datasets must have the same number and type of attributes.
2. The Java component requests to the CCN a Global Classifier. The CCN offers two options: to get the last global classifier stored in the Global Classifiers repository or to get a new one.

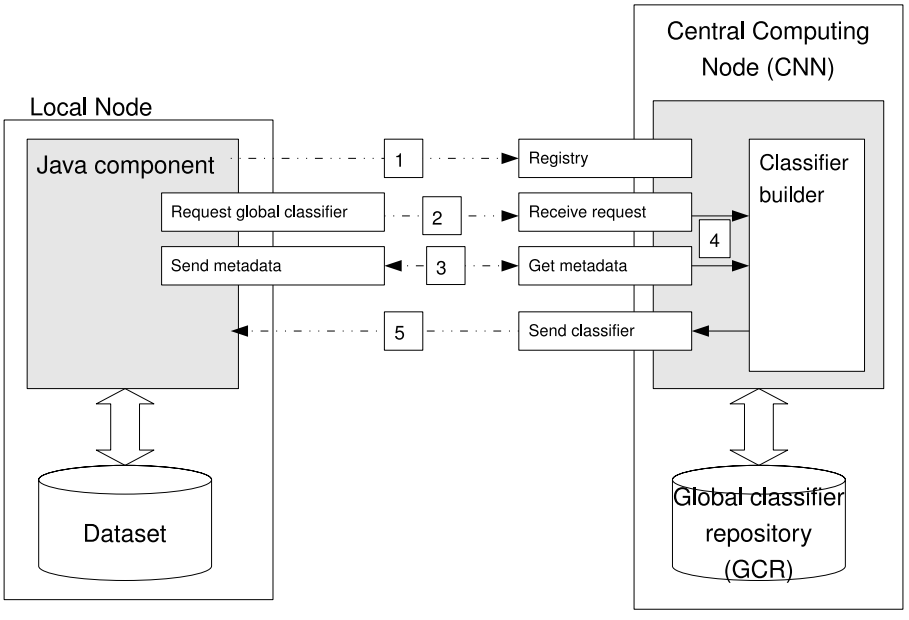


Fig. 2. Metadata interchange process

3. If the Java component chooses to built a new classifier then the CCN begins a number of iterations with all the registered Java components asking for metadata.
4. The metadata is sent to the CCN which interacts with all the local nodes asking for their metadata. This interactive process allows the CCN to built the Global Classifier.
5. Once the CCN has built a Global Classifier, it is saved and sent to the Java component that request for it.

4 ID3 Global Classifier

The Web application Team Miner implements this architecture which includes a classification algorithm for obtaining decision trees. The ID3 algorithm[6] has been adapted for our architecture. It is implemented using RMI and Web Services technologies. This algorithm works as follow:

- Classifier Builder asks metadata for each registered local node
- It creates a global group of metadata
- Invokes buildTree method sending global metadata as parameter
  - The buildTree method receives the metadata
  - Calculates the gain for each attribute
  - Chooses the attribute with best gain

- Appends the attribute to the tree
- If maxim gain is equal to zero
  - Gets a distribution of values of attributes
  - Normalizes the distribution
  - Sets the leaf's valueThis paper is organized as follows: Section 2 describes some of related work. Section 3 shows the architecture of our approach. Section 4 presents the result of the work done so far. Section 5 present the implementation. Finally, Section 5 concludes the paper with ongoing and future work.
- Else
  - Asks metadata for each registered local node
  - It creates a global group of metadata
  - The method invokes itself sending the new global metadata
- Return tree

– Finally it returns the decision tree

5 Implementation

This section gives a brief description of or Team Miner application, which is a Web application implemented using the architecture depicted in section 3. The first prototype of the Team Miner application implements the ID3 algorithm described in section 4. The basic data structures used in this implementation are next described.

1. **Remote interface**
   Code below shows the remote interface of Java component. It exposes two methods: *getAttributes*, it sends the attributes contained into the local sources, and *getMetadata*, it sends metadata wrapped into a Vector

```
public interface localInterface extends Remote{
public Vector getAttributes() throws RemoteException;
public Vector getMetadata() throws RemoteException;
public Vector getMetadata(int attribute, String value,
    int level, boolean leaf) throws RemoteException;
}
```

   This interface is implemented to interacting between the CCN and the local nodes
2. **Structure of metadata**
   Metadata are a summary of the real data obtained from structures of nested vectors. The Metadata Vector contains holds one vector for each attribute from the original source. Each attribute vector has one vector for each possible value that it can take. Then, each value vector contains the total number of instances belonging to every target class. An example of this structure is showed in Fig. 3.

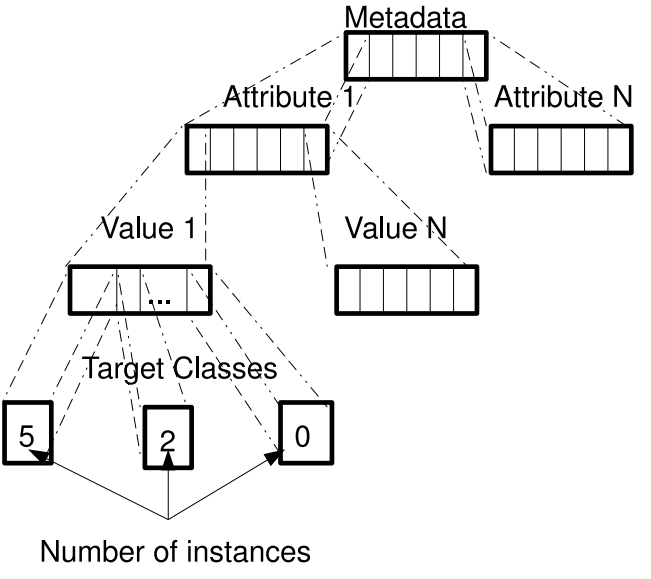


Fig. 3. Structure of metadata

3. **Central Node**
   CCN gathers metadata from all remote Java components, grouping them into a global metadata group. Then, the Classifier Builder starts to build the classifier by means of getting the most significant attribute and requiring more metadata. Finally, when it gets the classifier, the CCN sends it to the requester Java component. The mainly algorithms are shown in Algorithm 5.1 and Algorithm 5.2:

Algorithm 5.1: CLASSIFIER BUILDER ALGORITHM()

Classifier Builder
for each *Java component*
do
{*getMetadata*
end for
createGlobalMetadata
tree ← BUILDTREE(*GlobalMetadata*)
return (*tree*)
end Classifier Builder

Algorithm 5.2: BUILDTREE ALGORITHM(*GlobalMetadata*)

```
buildTree
{ getAttributesGain
  tree ← attribute with best gain
  if maxGain = 0
    then { get distribution of values of target class
           get distribution normalized
           set value of leaf
    else { for each value in attribute
           do
           { for each Java component
             do
             {getMetadata
             createGlobalMetadata
             tree ← BUILDTREE(GlobalMetadata)
  end if
  return (tree)
end buildTree
```

The algorithm 5.1 is implemented in CCN and is the initial step to build the classifier. It requests metadata from each Java component and after that a global metadata is built from them. Then it calls the build_tree function sending as parameter the group of global metadata. Finally, it obtains the global classifier or decision tree.

The algorithm 5.2 is the responsible of building the classifier. It receives the global metadata and then it calculates the gain of each attribute from the group. Once it has done this process it chooses the attribute with best gain and appends it to the tree. If the maxim gain is equal to zero means that it has found a leaf of the tree. In this case, the algorithm obtains the distribution of values of the target class, normalizes them and sets the value of the new leaf. Otherwise, it must split the group of global metadata in a set of subgroups. This action generates new branches for the tree; one for each value of the chosen

Table 1. Preliminary results

Classifier type	Training set	Percent of instances		
		Correctly Classified	Incorrectly Classified	Unclassified
Local	874	73.2	14.8	12
Local	1750	83.07	0	16.93
Local	4750	86.13	0.67	13.2
Global	7374	100	0	0

attribute. In order to do this process, it asks every local node for a new group of metadata according to the respective value. After that, it creates a new group of metadata and recurrently calls itself sending, as parameter, this new group. When this algorithm reaches a leaf it goes back looking for next subgroup of metadata. Finally, this algorithm returns the generated tree (the final global classifier).

6 Preliminary Results

It has been developed a DDM system that allow different users to register their dataset working in an interactive way for building a global classifier using the ID3 algorithm. It was built a test scenario employing breast cancer datasets taken from three different sites where each site emulates a system acceding interactively from a different hospital. The preliminary results are shown in table 1.

First column shows the classifier type. The size of training set(number of instances) is shown in the second column. Next 3 to 5 columns show the percent of correctly, incorrectly and unclassified instances respectively. Last row shows the Global Classifier results. All classifiers were tested using a test set of 750 instances.

Preliminary results show that using each individual dataset generates a classifier that offers between 70 to 85% of correctly classified instances while using the global one gets around 85 to 100% of correctly classified instances. These results support the fact that if we have a global vision of data the results will be closer to reality.

7 Conclusions

Data classifiers are tools that allow users to predict events based on historical information. Some institutions, such as hospitals, can obtain benefits from data classifiers, specially for illness prevention. However, many of them do not have enough historical information to built a good data classifier. Institutions with similar interest, like health care, would like to exchange information without lost data privacy looking for building data classifiers which help in tasks like illness prevention and developing of better clinical guides. Data classifiers performs better when they are built from global knowledge. This situation was the motivation for developing the Team Miner Web application based on a distributed data mining architecture which keeps privacy of data and take advantage of global knowledge. As ongoing and future work, this prototype is going to extend other popular classification and clustering algorithms.

Acknowledgments. This research was partially funded by project number 51623 from “Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas”.

References

1. Witten, H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann Publishers, San Francisco (2005)
2. Talia, D., Trunfio, P., Verta, O.: Weka4WS: A WSRF-Enabled Weka Toolkit for Distributed Data Mining on Grids. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 309–320. Springer, Heidelberg (2005)
3. Khoussainov, R., Zuo, X., Kushmerick, N.: Grid-enabled Weka: A Toolkit for Machine Learning on the Grid. ERCIM 59, 47–48 (2004)
4. Shaikh Ali, A., Rana, O.F., Taylor, I.J.: Web Services Composition for Distributed Data Mining. In: International Conference Workshop on Parallel Processing, pp. 11–18. IEEE, Los Alamitos (2005)
5. Peña, J.M., Sánchez, A., Robles, V., Pérez, M.S., Herrero, P.: Adapting the Weka Data Mining Toolkit to a Grid Based Environment. In: Szczepaniak, P.S., Kacprzyk, J., Niewiadomski, A. (eds.) AWIC 2005. LNCS (LNAI), vol. 3528, pp. 492–497. Springer, Heidelberg (2005)
6. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1(1), 81–106 (1986)
7. Ross Quinlan, J.: C4.5: programs for machine learning. Morgan Kaufmann, San Francisco (1993)
8. McQueen, J.: Some methods for classification and analysis of multivariations. In: Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
9. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid Prototyping for Complex Data Mining Tasks. In: 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2006)
10. University of Illinois and Data Mining Research Group and DAIS Research Laboratory, IlliMine 1.1.0, http://illimine.cs.uiuc.edu/
11. Statistics Department of the University of Auckland, R Project 2.6.1, http://www.r-project.org/
12. Williams, G.: Rattle 2.2.74, http://rattle.togaware.com/
13. Artificial Intelligence Unit of University of Dortmund, Yale 4.0, http://rapid-i.com/

Undecimated Wavelet Based Autoregressive Model for Anchovy Catches Forecasting

Nibaldo Rodriguez[1], Carlos Castro[2,⋆], Orlando Duran[1], and Eleuterio Yañez[1]

[1] Pontificia Universidad Católica de Valparaíso, Chile
FirstName.Name@ucv.cl
[2] Universidad Técnica Federico Santa María, Valparaíso, Chile
FirstName.Name@inf.utfsm.cl

Abstract. The aim of this paper is to find a model to forecast 1-month ahead monthly anchovy catches using un-decimated multi-scale stationary wavelet transform (USWT) combined with linear autoregressive (AR) method. The original monthly anchovy catches are decomposed into various sub-series employing USWT and then appropriate sub-series are used as inputs to the multi-scale autoregressive (MAR) model. The MAR's parameters are estimated using the regularized least squares (RLS) method. RLS based forecasting performance was evaluated using determination coefficient and shown that a 99% of the explained variance was captured with a reduced parsimony and high accuracy.

Keywords: forecasting, stationary wavelet transform, autoregressive.

1 Introduction

In fisheries management policy the main goal is to establish the future catch per unit of effort (CPUE) values in a concrete area during a known period keeping the stock replacements. To achieve this aim lineal regression methodology has been successful in describing and forecasting the fishery dynamics of a wide variety of species [1,2]. However, this technique is inefficient for capturing both nonstationary and nonlinearities phenomena in anchovy catch forecasting time series. Recently there has been an increased interest in combining nonlinear techniques and wavelet theory to model complex relationship in nonstationary time series. Nonlinear model based on Neural networks have been used for forecasting model due to their ability to approximate a wide range of unknown nonlinear functions [3]. Gutierrez *et. al.* [4], propose a forecasting model of monthly anchovy catches based on a sigmoidal neural network, whose architecture is composed of an input layer of 6 nodes, two hidden layers having 15 nodes each layer, and a linear output layer of a single node. Some disadvantages of this architecture is its high parsimony as well as computational time cost during the estimation of linear and nonlinear weights. As shown in [4], when applying the Levenberg

⋆ The second author has been partially supported by the Chilean National Science Fund through the project FONDECYT 1070268.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 325–332, 2008.

Marquardt (LM) algorithm, the forecasting model achieves a determination coefficient of 82%. A better result of the determination coefficient can be achieved if sigmoidal neural network is substituted by a linear autoregressive model combined with discrete wavelet transform based on translation-invariant wavelet transform. Coifman and Donoho [5] introduced translation-invariant wavelet denoising algorithm based on the idea of cycle spinning, which is equivalent to denoising using the un-decimated discrete stationary wavelet transform (USWT) [6,7]. Beside, Coifman and Donoho showed that USWT denoising achieves better root mean squared error than traditional discrete wavelet transform denoising. Therefore, we employ the USWT for monthly anchovy catches data preprocessing.

In this paper, we propose a linear autoregressive model combined with un-decimated stationary wavelet transform to forecast 1-month ahead monthly anchovy catches. The anchovy catches data considered is decomposed into some subseries by using USWT. Then one multi-scale autoregressive model is constructed with appropriate wavelet subseries as inputs. The MAR's parameters are estimated using the regularized least square method.

The rest of the paper is organized as follows: the hybrid wavelet-autoregressive model and estimating method are presented in Section 2. Section 3 presents some experiments and results related to forecasting monthly anchovy catches on the north area of Chile. Finally, the conclusions are drawn in the last section.

2 Wavelet and Autoregression Based Forecasting

This section presents the proposed forecasting model for monthly anchovy catches in north area of Chile, which is based on un-decimated stationary wavelet transform and linear autoregressive model. Moreover, instead of using the original data of past observations to predict $x(t+1)$, we will use its wavelet coefficients.

2.1 Stationary Wavelet Preprocessing

A signal can be represented at multiple resolutions by decomposing the signal on a family of wavelets and scaling functions [5,6,7]. The scaled (approximation) signals are computed by projecting the original signal on a set of orthogonal scaling functions of the form

$$\phi_{jk}(t) = \sqrt{2^{-j}}\phi(2^{-j}t - k) \tag{1}$$

or equivalently by filtering the signal using a low pass filter of length r, $h = [h_1, h_2, ..., h_r]$, derived from the scaling functions. On the other hand, the detail signals are computed by projecting the singal on a set of wavelet basis functions of the form

$$\psi_{jk}(t) = \sqrt{2^{-j}}\psi(2^{-j}t - k) \tag{2}$$

or equivalently by filtering the signal using a high pass filter of length r, $g = [g_1, g_2, ..., g_r]$, derived from the wavelet basis functions. Finally, repeating the

decomposing process to any scale J, the original signal can be represented as the sum of all detail coefficients and the last approximation coefficient.

In time series analysis, discrete wavelet transform (DWT) often suffers from a lack of translation invariance. This problem can be tacked by mean of the un-decimated stationary wavelet transform. The USWT is similar to the DWT in that the high-pass and low-pass filters are applied to the input signal at each level, but the output signal is never decimated. Instead, the filters are upsampled at each level.

Consider the following discrete signal $x(n)$ of length N where $N = 2^J$ for some integer J. At the first level of USWT, the input signal $x(n)$ is convolved with h filter to obtain the approximation coefficients $a(n)$ and with g filter to obtain the detail coefficients $d(n)$, so that

$$a(n) = \sum_k h(n-k)x(k) \tag{3a}$$

$$d(n) = \sum_k g(n-k)x(k) \tag{3b}$$

Because no subsampling is performed, $a(n)$ and $d(n)$ are of length N instead of $N/2$ as in the DWT case. The output of the USWT is then the approximation coefficients and the detail coefficients $[a(n)\ d(n)]$. In this paper, the $h(n)$ and $g(n)$ filters are based on Haar wavelet filter and are given as

$$h = \left[\frac{1}{\sqrt{2}}\ \frac{1}{\sqrt{2}}\right] \tag{4a}$$

$$g = \left[\frac{-1}{\sqrt{2}}\ \frac{1}{\sqrt{2}}\right] \tag{4b}$$

2.2 Multiscale Autoregressive Forecaster

In order to predict the future signal $x(n+1)$ we can separate original signal $x(n)$ into two component as it is mentioned in (3). The first component $a(n)$ presents the tendency of the series and is characterized by slow dynamics $y(n)$, and second component $d(n)$ presents stochastic part of the series and characterized by fast dynamics $z(n)$. Therefore our forecasting model will be the co-addition of two predicted values given as

$$x(t+1) = y(t+1) + z(t+1) \tag{5}$$

The component $z(n)$ is estimates by using one multi-scale autoregressive (MAR) model given as

$$z(n+1) = \sum_{j=1}^{J} \sum_{i=1}^{p} \alpha_{j,i} d_j[t-(i-1)] \tag{6}$$

where p represents the order of the MAR(p,J) model and the J value denotes the multi-scale level of the stationary wavelet transform.

The component $y(n)$ is estimated by using one autoregressive (AR) model, which is given as

$$y(n+1) = \sum_{i=1}^{p} \beta_i a[n-(i-1)] \tag{7}$$

In order to estimate the linear parameters $\{\alpha_{j,i}\}$ or $\{\beta_i\}$ the regularized least square method is proposed. Now suppose a set of N_s training input-output samples, then we can perform N_s equations of the form of (6) or (7) as follows

$$Z = \alpha\Phi \tag{8a}$$
$$Y = \beta\Psi \tag{8b}$$

where

$$\Phi = [d_j(n), d_2(n-1), \cdots, d_j(n-(p-1))] \tag{9a}$$
$$\Psi = [a(n), a(n-1), \cdots, a(n-(p-1))] \tag{9b}$$

The optimal values of the linear parameters $\alpha_{j,i}$ or β_i are obtained using the following error objective function

$$E = \sum_{i=1}^{N_s} \left[H(n+1) - z(n+1)\right]^2 + \gamma \parallel \alpha \parallel^2 \tag{10a}$$
$$H(n+1) = \sum_{j=1}^{J} d_j(n+1) \tag{10b}$$
$$E = \sum_{i=1}^{N_s} \left[L(n+1) - y(n+1)\right]^2 + \gamma \parallel \beta \parallel^2 \tag{10c}$$
$$L(n+1) = a(n+1) \tag{10d}$$

where $\|\cdot\|$ denotes the eucludian norm and γ is a regularization parameter.

Minimizing the error objective function results is

$$\alpha = (\Phi^T\Phi + \gamma I)^{\dagger}\Phi^T Z \tag{11a}$$
$$\beta = (\Psi^T\Psi + \gamma I)^{\dagger}\Psi^T Y \tag{11b}$$

where I is the identity matrix and $(\cdot)^{\dagger}$ is the Moore-Penrose generalized matrix inverse [8].

3 Experiments and Results

The monthly anchovy catches data was conformed by historical data from January 1963 to December 2005, divided into two data subsets as shown in Fig.1. In the first subset, the data from January 1963 to December 1985 was chosen for the training phase (weights estimation), while the remaining was used

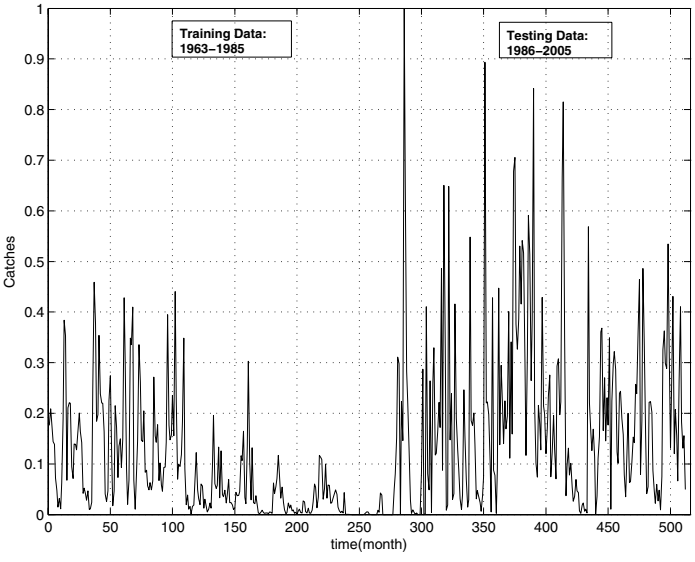


Fig. 1. Observed monthly anchovy catches data

Table 1. Statistic of the forecasting model with RLS and $\gamma = 0.003$

MAR(1,J) Model	MSE	R-Squared
$J = 1$	0.008823	76.00
$J = 2$	0.001487	96.12
$J = 3$	0.000598	98.27
$J = 4$	**0.000140**	**99.60**

Table 2. Statistic of the forecasting model with RLS and $\gamma = 0.003$

MAR(2,J) Model	MSE	R-Squared
$J = 1$	0.009324	74.51
$J = 2$	0.001600	95.69
$J = 3$	0.000529	98.50
$J = 4$	**0.000122**	**99.70**

for the validation phase. The forecasting process starts by applying the undecimated stationary wavelet transform and normalization step to the original anchovy catches data. Then, the RLS method are performed to adapt the MAR's parameters with regularization parameter γ. In order to estimate the regularization parameter we apply the trail and error method and was selected as $\gamma = 0.003$.

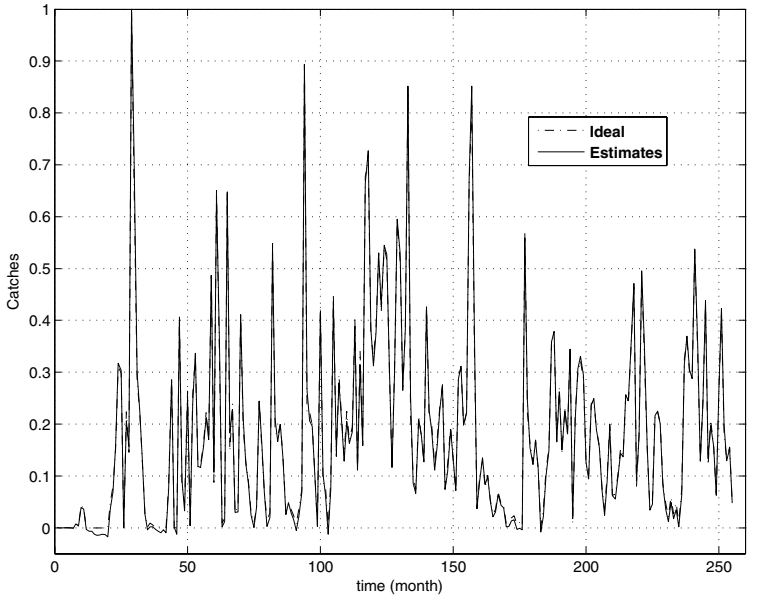


Fig. 2. Observed anchovy catches vs estimated anchovy catches with test monthly data

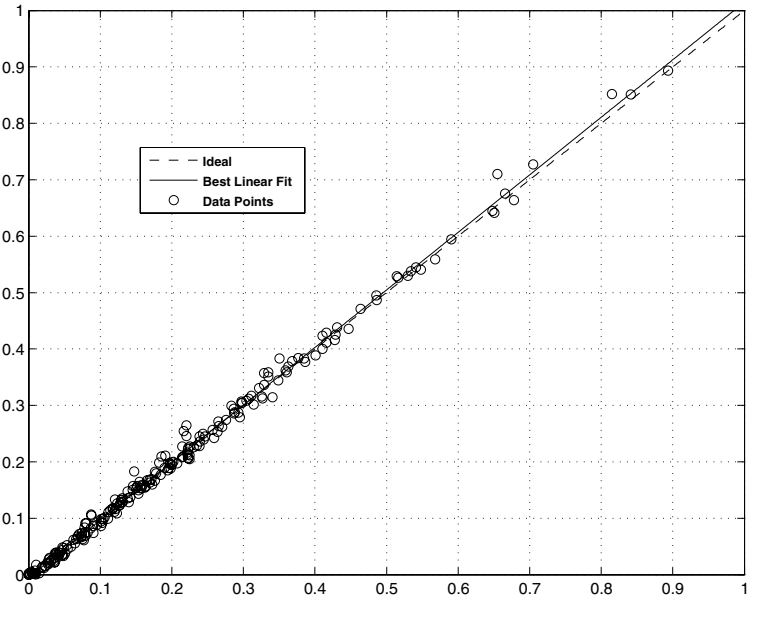


Fig. 3. Observed anchovy catches

The proposed forecaster performance according to fitness function (MSE) and determination coefficient R^2 are given in Table 1 and Table 2. From Tables 1 and 2, we conclude that MAR forecasting model based on 4-level wavelet with lagged two-values, when compared with MAR forecasting model based on 4-level wavelet and lagged one-value; finds a better determination coefficient (99.70%) using $8 + 4 = 12$ linear parameters.

Fig.2 describes the performance evaluation of the validation phase with testing data for the MAR(2,4) forecasting model. This plot shows the original time series and the one predicted by proposed forecasting model. The difference is so tiny that it is impossible to tell one from another by eye inspection. That is why you probably see only the MAR(2,4) forecasting curve. Besides, from Fig.2 it can be observed that the best forecasting model according to its parsimony and precision is the architecture composed by two lagged values and 4-level wavelet.

Fig.3 illustrates the determination coefficient estimation of the validation phase between the observed and estimated anchovy catches data with the best forecasting model. Please note that the forecasting model shown that a 99.70% of the explained variance was captured by the proposed MAR(2,4) model.

4 Conclusions

A un-decimated stationary wavelet autoregressive model with regularized least square method for 1-monthly ahead monthly anchovy catches forecasting has been presented. Firstly, monthly anchovy catches data was decomposed into sub-series by un-decimated stationary wavelet transform. Then appropriate sub-series were used as input to the forecasting model. The results found shown that a 99.70% of the explained variance was captured with a reduced parsimony and high speed convergence.

References

1. Stergiou, K.I.: Prediction of the Mullidae fishery in the easterm Mediterranean 24 months in advance. Fish. Res. 9, 67–74 (1996)
2. Stergiou, K.I., Christou, E.D.: Modelling and forecasting annual fisheries catches: comparison of regression, univariate and multivariate time series methods. Fish. Res. 25, 105–138 (1996)
3. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Network 2(5), 359–366 (1989)
4. Gutierrez, J.C., Silva, C., Yañez, E., Rodriguez, N., Pulido, I.: Monthly catch forecasting of anchovy engraulis ringens in the north area of Chile: Nonlinear univariate approach. Fisheries Research 86, 188–200 (2007)
5. Coifman, R.R., Donoho, D.L.: Translation-invariant denoising, Wavelets and Statistics. Springer Lecture Notes in Statistics, vol. 103, pp. 125–150. Springer, Heidelberg (1995)

6. Nason, G., Silverman, B.: The stationary wavelet transform and some statistical applications, Wavelets and Statistics. Springer Lecture Notes in Statistics, vol. 103, pp. 281–300. Springer, Heidelberg (1995)
7. Pesquet, J.-C., Krim, H., Carfantan, H.: Time-invariant orthonormal wavelet representations. IEEE Trans. on Signal Processing 44(8), 1964–1970 (1996)
8. Serre, D.: Matrices: Theory and applications. Springer, New York (2002)

Neural Processor as a Dynamic Power Manager for Digital Systems

Adam Gołda and Andrzej Kos

AGH University of Science and Technology, Department of Electronics,
30 Mickiewicza Ave,
30-059 Kraków, Poland
{golda,kos}@agh.edu.pl

Abstract. In this paper the utilization of neural processors as supervisory units that control predictive techniques of dynamic power management is described. Power management becomes more and more important as density of power dissipated in modern integrated circuits, especially microprocessors, continuously raises and can be even higher than 4 megawatts per square meter. It causes temperature increases that might be dangerous for the chip. The presented supervisors that are based on neurons allow correct prediction of chip temperature on the basis of current temperature, power losses that will be consumed in the next units of time, as well as previous power dissipations. Their task is to keep the throughput of high-frequency and high-efficiency systems on the highest possible level under the conditions of energy savings and maintaining safe temperature of chip. The supervisory units are designed using 32-bit and fixed-point precision.

Keywords: Neural networks, Dynamic power management, VLSI systems.

1 Introduction

The thermal aspect has become a very significant matter for integrated circuits designed and produced in latest *CMOS* technologies. The density of power dissipated in semiconductor structures rises to significant values, even more than $4.2 \cdot 10^6$ W/m^2 [1], and because of that die temperature could achieve high values that can be even dangerous for chip. As a result, modern and potential upcoming high-efficiency integrated circuits fabricated in deep-sub-micrometer and nano-meter technologies are supposed to be controlled with respect to temperature [2], i.e. the temperature of chip cannot exceed the maximum allowed value denoted as T_{al_max}. Thermal control of integrated circuits utilizes the following dynamic power management (*DPM*) techniques: dynamic clock throttling (*DCT*), dynamic frequency scaling (*DFS*), and dynamic voltage scaling (*DVS*) [2] – [5]. These methods are used to realize two main aims:

-to decrease the power consumption of battery-operated mobile devices [3], [6] – [10]. The frequency and supply voltage of mobile device have theirs maximum values only for compute-intensive and short-latency processes [3]. Background and long-latency tasks might be performed slower thus the frequency and supply voltage could

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 333–342, 2008.

be adequately decreased. The same is done for idle states of system. In other words, the values of supply voltage and frequency are adapted to the application execution time and its deadline, e.g. [8], [10].

-and to control the temperature of chip [2], [11]. In the high efficiency systems, the chip temperature is the deciding factor in switching the state of full or reduced performance of system. The die temperature is indirectly the result of performed action. Therefore, microprocessor devices, which use such dynamic techniques, require an operating system that is able to predict the requirements of individual process and control the frequency and supply voltage, for instance using Advanced Configuration and Power Interface *ACPI*, [3], [11]. It is also possible to control performance using hardware mechanisms [2].

The latter purpose can be extended and utilized to keep the throughput of high efficiency systems on the possible maximum level under the condition of energy savings [12], [13] and this method is taken into consideration in this paper. The neuron-based supervisory units, that follow the current temperature, power consumption, and future requirements for power, are elaborated in this work.

As far as the temperature, which is the result of power density, is concerned, power losses of *CMOS* circuits should be shortly introduced. In this work the terms *energy* and *power* are often used interchangeably especially when the general matter of consumption i.e. losses is discussed. Obviously, conversion between these quantities might be easily performed.

The whole amount of energy consumed by *CMOS* circuits E_{tot} can be divided into two main components: a static E_{st} and a dynamic E_{dyn}. The dynamic energy consist of a short-circuit E_{dynSC} and a capacitance charging/discharging E_{dynCAP} parts, e.g. [14] – [17].

There are several leakage mechanisms i.e. sources of static losses [15], [18], [19] that might be categorized into three groups connected with directions of current flows in solid of *MOS* transistors [20], [21]:

- flow between sources and drains: sub-threshold and channel punchthrough currents,
- flow from gate to bulk: oxide tunneling leakage and current due to hot-carrier injection,
- flow through *pn*-junction structures: reverse bias *pn* junction, band-to-band tunneling *BTBT*, and gate inducted drain leakage *GIDL* currents.

The dynamic components appear during gate switching. Current that flows in order to reload the charge collected by capacitors summed as total load capacitance C_L causes the dynamic capacitance charging/discharging energy consumption E_{dynCAP}, which is also called the capacitive energy. Non-zero rising and falling times of input signal cause I_{dynSC} current flow from supply source through series connection of *NMOS* and *PMOS* transistors to the ground. The short-circuit energy E_{dynSC} is associated with such current.

Describing it in abridgement, the pure dynamic power management techniques cannot decrease static power consumption but they are invaluable for reduction of dynamic parts. They are proportional to switching frequency, and are dependent on supply voltage in square (capacitive one) or even cubic matter (quasi short-circuit one) [1].

2 Principles of *DPM* Techniques

DCT and *DFS* techniques are based on frequency control. Theirs application in power management of integrated circuits are grounded in clock shutdowns (*DCT*) or frequency reductions (*DFS*). Both of them cause the diminution of the power consumed by system. *DVS* method is very often combined with *DFS* technique. In this situation the capacitance dynamic power consumption is reduced in a cubic manner [2], [9] in accordance with the following formula

$$P_{dynCAP} = f \cdot C_L \cdot V_{DD}^2 \ , \tag{1}$$

where P_{dynCAP} is the dynamic part of power consumption of *CMOS* gates, f – effective clock frequency, C_L – load capacitance, and V_{DD} – supply voltage.

DCT and *DFS* techniques reduce dynamic power consumption, but do not decrease this part of energy per process because it is independent of clock frequency. This can be done only by supply voltage lowering *DVS* [3].

Let us illustrate the performance and principles of operations of dynamic power management techniques.

The thermal behavior of the chip is modeled by means of the first order system. It is assumed that thermal resistance and thermal capacitance have known values. Thus the thermal time constant τ as well as the steady and unsteady states temperatures at any point of the die might be calculated. In addition, the die temperature is assumed to be identical on the whole bulk surface.

All the considered common methods of dynamic power management need the high and low threshold temperatures that identify when to switch between the modes of reduced and full performance, Fig. 1. The problem of proper selection of values of these thresholds is deeply described in [1], [2], [12], [13], and it is not raised in this paper.

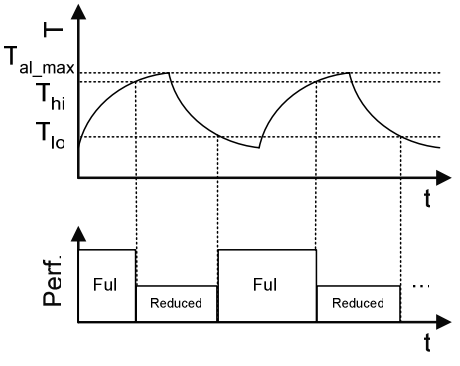


Fig. 1. Chip performance versus temperature alteration

Obviously, the same job is done faster in full performance mode than in reduced one. Thus, it is assumed that whole job is divided into parts. One part of this job is done during one unit of time. This piece of time is fixed, e.g. it lasts several or several hundred cycles of clock in full performance mode and adequately greater in reduced

mode. The values of duration of these work units have to be set separately for specific digital system. It is also evident that toggling between modes cannot be done before the end of currently performed part.

Let us make also some more assumptions about power consumption and other parameters of the analyzed digital systems. All analyses and simulations are performed for normalized quantities. The digital system has task, which consists of 200 parts.

The power dissipations are set to simulations randomly (uniform probability distribution) for certain periods of time. The power losses are assumed to be at level of 0.3 for static dissipation and from 0 to 0.7 for dynamic consumption. These values are valid for normal i.e. full speed work of system. In the reduced mode, i.e. for decreased frequency and/or voltage, the dynamic power losses are properly smaller, [1], [12], [13]. When clock is turned off chip consumes only static energy. The assumption that static and dynamic power losses have strictly established values (i.e. they are independent of temperature) can be made without additional constraints because as we will see the chip temperature oscillates between two threshold temperatures that are close together. Additionally, in these deliberations the temperature T_{al_max} is set to 0.55.

The behavior of digital system controlled by means of common *DVS* method is presented in Fig. 2.

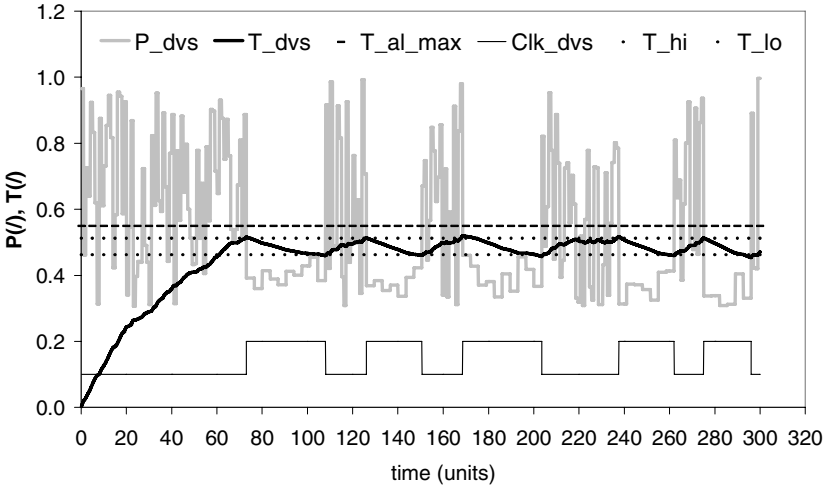


Fig. 2. Transient temperature response of system; the random power losses; system with common *DVS*

This figure depicts also the clock state, denoted as *Clk_dvs*, which magnitude is presented as 0.1 or 0.2 instead of 1 and 2 in order to obtain better visible diagram. The clock works with its maximum frequency for the state equaled 'one'. The state 'two' stands for reduced frequency (*DFS*) or reduced frequency and supply voltage (*DVS*).

The execution of total job (200 work units) lasts 300 total units of time for system controlled by *DVS*. The energy consumption equals about 156.94 units (energy is also

normalized that is why it is presented in units instead of Joules). This quantity has such big value because it is a sum of energy losses in particular time units.

3 Predictive *DPM* Techniques

Allowing for temperature the predictive algorithms of dynamic control of system performance boil down to the fact that after exceeding the high threshold T_{hi} the temperature may not reach the allowed maximum limit T_{al_max}. For that reason, the system can carry on its work on the condition that the power consumption in the next work unit of time will not overheat the chip. Moreover, it is possible that the subsequent power losses will be smaller and the semiconductor structure will cool down. The high threshold temperature approaches the T_{al_max} for ideal predictive *DCT*-, *DFS*-, and *DVS*-based methods: $T_{hi} \rightarrow T_{al\_max}$. Predictive techniques require information, e.g. from operating system, about the value of power consumption in the nearest future. During these analyses the term *nearest future* means two time units and is denoted as Δt.

Fig. 3 depicts the results obtained for system controlled by ideal predictive *DVS* technique that is denoted as *VPR*.

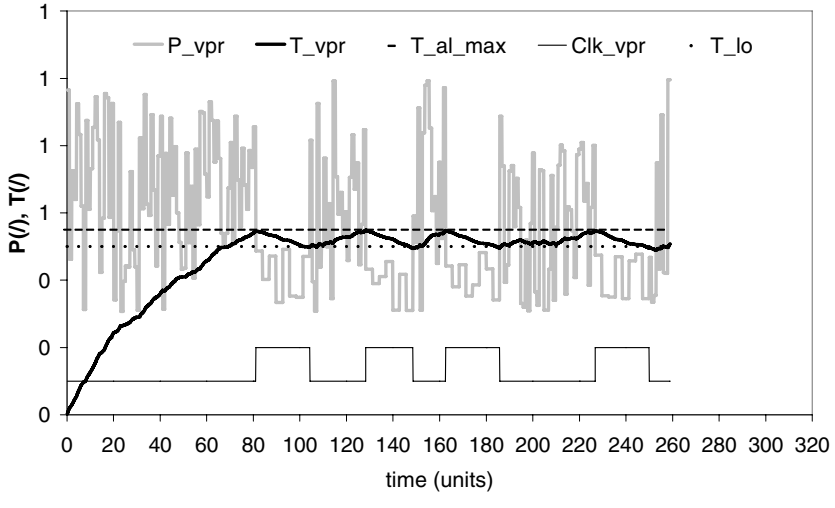


Fig. 3. Transient temperature response of system; the random power losses; system controlled by ideal predictive *DVS* i.e. *VPR* technique

The predictive techniques allow better performance than common ones. The work is performed faster - 258.9 time units, the profit is about 13.7 %. The yield is also made in energy consumption, which is reduced by 7.6 % - total losses are about 144.98 units.

The theoretical results obtained for ideal predictive dynamic techniques are calculated as a sum of the responses to the unit step functions of the power dissipated in particular work units. Thus, the forecast temperature is the outcome of the whole previous dissipations, current temperature, and the power that will be consumed in the next units of time. Obviously, the real supervisor cannot accumulate all values of earlier consumption. The works [1] and [13] introduce the supervisory units that predict the temperature on the basis of the current temperature, power consumed in the next two time units, and previous dissipations. The behavior of the simplest one, which has no memory i.e. does not take into account previous power losses, is described by

$$0 = T_0 \cdot \left(1 - e^{-\frac{\Delta t}{2\cdot\tau}} + e^{-\frac{3\cdot\Delta t}{2\cdot\tau}}\right) + \Delta T_{1.ss} \cdot \left(1 - e^{-\frac{\Delta t}{\tau}}\right) + \Delta T_{2.ss} \cdot \left(1 - e^{-\frac{\Delta t}{2\cdot\tau}}\right) + T_{100} \cdot \left(e^{-\frac{\Delta t}{2\cdot\tau}} - e^{-\frac{3\cdot\Delta t}{2\cdot\tau}}\right) - T_{al\_\max} \qquad (2)$$

where T_0 – current temperature, T_{100} – steady state temperature obtained for the maximum power dissipated in the chip, τ – thermal time constant, $\Delta T_{1.ss}$ – the steady state growth of temperature obtained for the power step in the first next time unit, $\Delta T_{2.ss}$ – the steady state growth of temperature obtained for the power step in the second next time unit, Δt – two time units.

The shape of this equation allows direct implementation in neural network, see Fig. 4. The free term in this expression is the weight of bias. Clearly, considered supervisors can also be easily applied in software or other hardware.

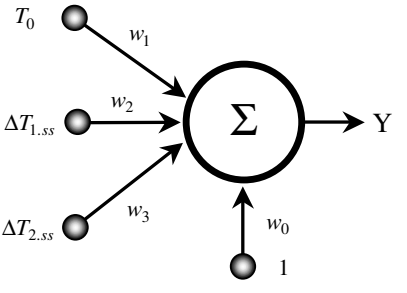


Fig. 4. Neuron that works as supervisor

On the basis of the formula (2) the supervisor has been designed using *VHDL* and logic synthesis in *Cadence* environment. Fig. 5 presents a part of the source code and attained topography of this 32-bit supervisory unit in 0.35 μm technology.

The operation of this supervisor is explained in Fig. 6, and Fig. 7 presents the results obtained for system controlled by this supervisory unit, this realization is denoted as *VN1*. This supervisory unit, i.e. neuron, predicts the performance mode of the digital system. For example, in time t_0 it decides about this state in time t_1 on the basis of current temperature and the power that will be consumed in the next two time units (see Fig. 6).

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity neuron_3_in is
.       port(
.       .       clk : in STD_LOGIC;
.       .       input1 : in STD_LOGIC_VECTOR(31 downto 0);
.       .       input1_valid : in STD_LOGIC;
.       .       scale1 : in STD_LOGIC_VECTOR(31 downto 0);
.       .       input2 : in STD_LOGIC_VECTOR(31 downto 0);
.       .       input2_valid : in STD_LOGIC;
.       .       scale2 : in STD_LOGIC_VECTOR(31 downto 0);
.       .       input3 : in STD_LOGIC_VECTOR(31 downto 0);
.       .       input3_valid : in STD_LOGIC;
.       .       scale3 : in STD_LOGIC_VECTOR(31 downto 0);
.       .       bias : in STD_LOGIC_VECTOR(31 downto 0);
.       .       output : out STD_LOGIC_VECTOR(31 downto 0);
.       .       data_ready : out STD_LOGIC
.           );
end neuron_3_in;
```

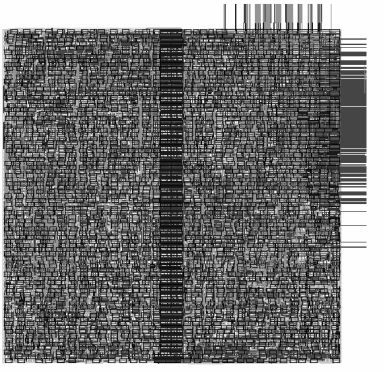


Fig. 5. A part of the source code of the considered neuron that works as supervisor (left), and its topography (right)

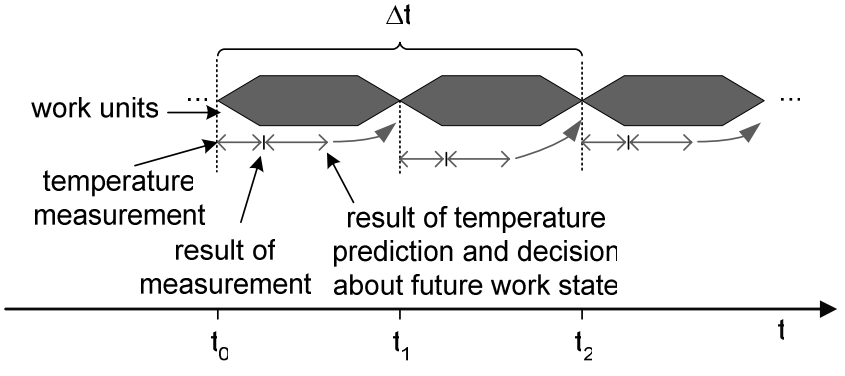


Fig. 6. Operation of supervisor

As expected, performance of such system is worse than performance of system with ideal prediction, but better than common *DVS*-based one. It does its whole work in 277 time units (profit 7.66 %). The total energy consumption is at the level of 149.48 and is smaller of about 4.75 % than for common *DVS* technique.

Let us illustrate the behavior of supervisory unit on the example of situation before the 80th time unit – the first change of performance mode, Fig. 7. Fig. 8 presents the results obtained from the considered 32-bit supervisor for two sequential steps. The first one does not require the performance reduction yet, but the second one switches on the reduced mode. As we can see, the output is positive for the first case and negative for the second one (the end of calculation of performance mode is indicated by positive slope of *data_ready* signal in the Fig. 8). Therefore we decided not to use the additional block with activation function but use the sign bit instead. It simplifies the circuit.

The designed supervisory unit works properly. Its 32-bit architecture is enough for this purpose.

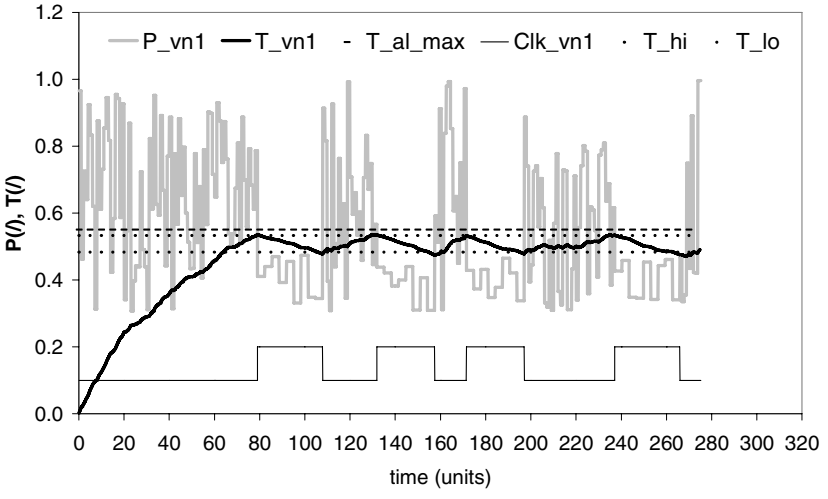


Fig. 7. Transient temperature response of system; the random power losses; system controlled by real predictive *DVS* technique, that is based on considered supervisor and denoted as *VN1*

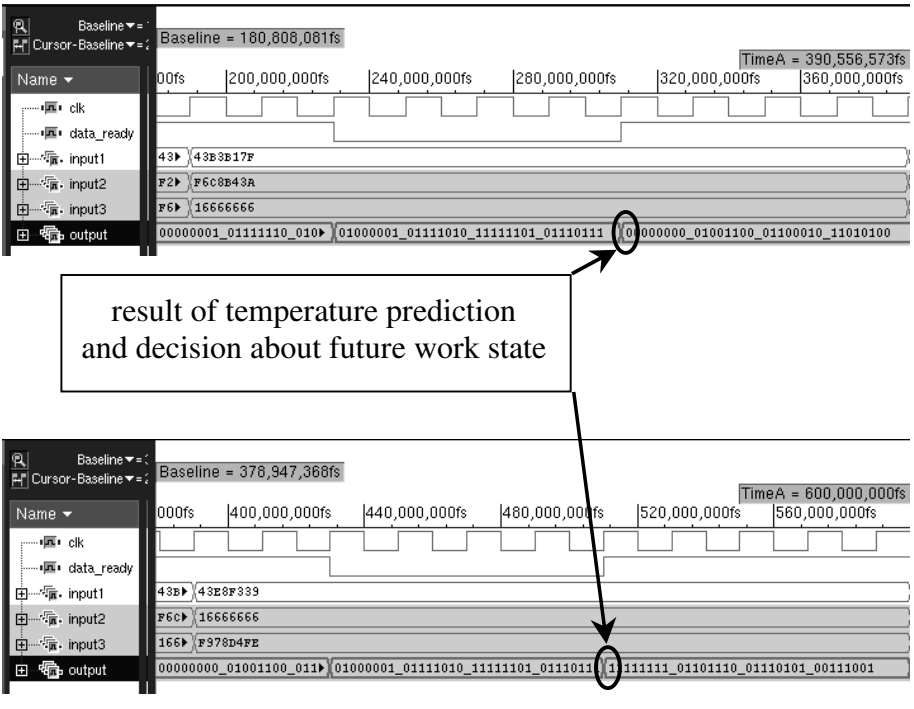


Fig. 8. Exemplary results of system mode prediction by considered 32-bit supervisory unit; decision of remaining in full performance mode (top), and switching into reduced mode (down)

4 Conclusions

The new predictive techniques that are based on common *DCT*, *DFS*, and *DVS* methods have been analyzed. They are dedicated to high efficiency systems in order to keep theirs throughput on the possible maximum level. They consist in cooperation of supervisory unit with operating system that has to deliver data about the current and future process requirements of power consumption.

The simply shape of equations, which describes the behavior of the supervisors, allows direct implementation in neural networks. The presented supervisors can also be easily applied in the other hardware or software.

The main aim of these units is energy reduction, which is improved up to about 4.75 % for the simplest predictor in *DVS*-based technique.

Not only the supervisory units turn a profit in energy consumption, they also gain in performance even to over 7.6 % for the simplest one.

The greater performance of the system is connected with the temperature increase. But the supervisory units ensure that the maximum allowed temperature in not crossed.

The analyses of energy consumption do not take into account energy losses of neuron circuit, thus the energy profits will be a little bit smaller in real applications.

References

1. Gołda, A.: Reduction of Energy Losses in CMOS Circuits Considering Die Temperature. Doctoral thesis (2008)
2. Daasch, W.R., Lim, C.H., Cai, G.: Design of VLSI CMOS Circuits Under Thermal Constraint. IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing 49(8), 589–593 (2002)
3. Burd, T.D., Pering, T.A., Stratakos, A.J., Brodersen, R.W.: A Dynamic Voltage Scaled Microprocessor System. IEEE J. of Solid-State Circuits 35(11), 1571–1580 (2000)
4. Nowka, K.J., et al.: A 32-bit PowerPC System-on-a-Chip With Support for Dynamic Voltage Scaling and Dynamic Frequency Scaling. IEEE J. of Solid-State Circuits 37(11), 1441–1447 (2002)
5. Kim, B.-G., Kim, L.-S.: A 250-MHz–2-GHz Wide-Range Delay-Locked Loop. IEEE J. of Solid-State Circuits 40(6), 1310–1321 (2005)
6. Pouwelse, J., Langendoen, K., Sips, H.: Dynamic Voltage Scaling on a Low-Power Microprocessor. In: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking MobiCom 2001, Rome, Italy, July 16-21, 2001, pp. 251–259 (2001)
7. Chowdhury, P., Chakrabarti, C.: Static Task-Scheduling Algorithms for Battery-Powered DVS Systems. IEEE Trans. on Very Large Scale Integration (VLSI) Systems 13(2), 226–237 (2005)
8. Hua, S., Qu, G.: Voltage Setup Problem for Embedded Systems with Multiple Voltages. IEEE Trans. on Very Large Scale Integration (VLSI) Systems 13(7), 869–872 (2005)
9. Zhai, B., Blaauw, D., Sylvester, D., Flautner, K.: The Limit of Dynamic Voltage Scaling and Insomniac Dynamic Voltage Scaling. IEEE Trans. on Very Large Scale Integration (VLSI) Systems 13(11), 1239–1252 (2005)

10. Chen, J.-J., Yang, C.-Y., Kuo, T.-W., Shih, C.-S.: Energy-Efficient Real-Time Task Scheduling in Microprocessor DVS Systems. In: Proceedings of the 12th Asia and South Pacific Design Automation Conference ASP-DAC 2007, Yokohama, Japan, January 23-26, 2007, pp. 342–349 (2007)
11. Zhang, Y., Chakrabarty, K.: A Unified Approach for Fault Tolerance and Dynamic Power Management in Fixed-Priority Real-Time Embedded Systems. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 25(1), 111–125 (2006)
12. Gołda, A., Kos, A.: Predictive frequency control for low power digital systems. In: Proceedings of the International Conference Mixed Design of Integrated Circuits and Systems MIXDES 2006, Gdynia, Poland, June 22-24, 2006, pp. 441–445 (2006)
13. Gołda, A., Kos, A.: Effective Supervisors for Predictive Methods of Dynamic Power Management. In: Proceedings of the 14th International Conference Mixed Design of Integrated Circuits and Systems MIXDES 2007, Ciechocinek, Poland, June 21-23, 2007, pp. 381–386 (2007)
14. Bhavnagarwala, A.J., Austin, B.L., Bowman, K.A., Meindl, J.D.: A minimum Total Power Methodology for Projecting Limits on CMOS GSI. IEEE Trans. on Very Large Scale Integration (VLSI) Systems 8(3), 235–251 (2000)
15. Wei, L., Roy, K., De, V.K.: Low Voltage Low Power CMOS Design Techniques for Deep Submicron ICs. In: Proceedings of the Thirteenth International Conference on VLSI Design, Calcutta, India, January 3-7, pp. 24–29 (2000)
16. Dongyan, H., Ming, Z., Wei, Z.: Design Methodology of CMOS Low Power. In: Proceedings of the IEEE International Conference on Industrial Technology IEEE ICIT 2005, Hong Kong, December 14-17, 2005, pp. 114–118 (2005)
17. Liao, W., He, L., Lepak, K.M.: Temperature and Supply Voltage Aware Performance and Power Modeling at Microarchitecture Level. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 24(7), 1042–1053 (2005)
18. Semenov, O., Vassighi, A., Sachdev, M.: Impact of technology scaling on thermal behavior of leakage current in sub-quarter micron MOSFETs: perspective of low temperature current testing. Microelectronics Journal 33, 985–994 (2002)
19. Roy, K., Mukhopadhyay, S., Mahmoodi-Meimand, H.: Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits. Proceedings of the IEEE 91(2), 305–327 (2003)
20. Abdollahi, A., Fallah, F., Pedram, M.: Leakage Current Reduction in Sequential Circuits by Modifying the Scan Chains. In: Proceedings of the Fourth International Symposium on Quality of Electronic Design ISQED 2003, San Jose, USA, March 24-26, 2003, pp. 49–54 (2003)
21. Kuźmicz, W., Piwowarska, E., Pfitzner, A., Kasprowicz, D.: Static Power Consumption in Nano-CMOS Cicruits: Physics and Modelling. In: Proceedings of the 14th International Conference Mixed Design of Integrated Circuits and Systems MIXDES 2007, Ciechocinek, Poland, June 21-23, 2007, pp. 163–168 (2007)

Tank Model Coupled with an Artificial Neural Network

Gustavo Cerda-Villafana[1], Sergio E. Ledesma-Orozco[1], and Efren Gonzalez-Ramirez[2]

[1] FIMEE, Universidad de Guanajuato,
Salamanca, Guanajuato, Mexico
Gustavo Cerda-Villafana, gcerdav@salamanca.ugto.mx
http://www.fimee.ugto.mx
[2] Universidad de Zacatecas,
Zacatecas, Zacatecas, Mexico

Abstract. Tank models have been used in many Asian countries for flood forecasting, reservoir operation, river basin modeling, etc. In this work a tank model is coupled with an ANN (Artificial Neural Network) for modeling a rainfall-runoff process. The ANN controls six of the tank model parameters to adjust them along time in order to improve efficiency. The data used in the simulations were collected from the Brue catchment in the South West of England. It should be pointed out that the raingauge network in this study is extremely dense (for research purposes) and does not represent the usual raingauge density in operational flood forecasting systems.

1 Introduction

The section of the hydrological cycle involving rainfall, basin characteristics, water flow, evaporation and water filtering is known as rainfall-runoff process. Many models have been developed to simulate rainfall-runoff processes mainly because they are nonlinear and time-variant. One of them, known as "tank model" was developed by Sugawara [1] and it has been used for flood forecasting, reservoir operation, river basin modeling, etc. The model has been modified and some mathematical tools have been adapted in order to improve its efficiency [2]. In this work an Artificial Neural Network (ANN) was adapted to the tank model to modify several of its parameters to make it "adaptable" to time variations.

The development of Artificial Neural Networks (ANNs) began approximately 50 years ago [3], trying to emulate neurons in the human brain. Mathematical descriptions can be consulted elsewhere, for example, [4], [5], and [6] present ANNs for general purposes. The ASCE Task committee [7] [8] published an excellent two-part series introduction to ANNs in hydrologic applications.

The ability of ANNs to associate input arguments to target values, independently of differences in dimension and magnitude, generates the expectation to discover the relationship between tank model parameters and rainfall estimations. To address this issue a model was developed using rainfall and runoff data from the "Natural Environment Research Council – HYdrological Radar EXperiment" (NERC HYREX) project.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 343–350, 2008.

2 Analysis

2.1 Tank Model

A tank model is a conceptual representation of a basin hydrological process. It simulates the "wetness" of the several soil layers by tanks arranged vertically in series and each one adapted with one or more outlets to account for water flow and filtration to lower layers. Figure 1 shows the model developed for this work.

The first tank is intended to simulate the basin surface down a few centimeters underneath. Its wetness is represented by the water level S1. The second tank is considered to cover the range from the first tank to several dozens of centimeters and the third thank covering the range containing groundwater affecting the basin.

In Fig. 1 A1, B1, and C1 represent the runoff coefficients of the first, second, and third tanks respectively. The unit of these coefficients is 1/hour. HA1, HB1, and HC1 are the heights of runoff orifices of each tank and their unit is mm. The infiltration coefficients are A0, B0, and C0, and their unit is 1/hour. Finally, S1, S2, and S3 reflect the physical concept of wetness and their unit is mm.

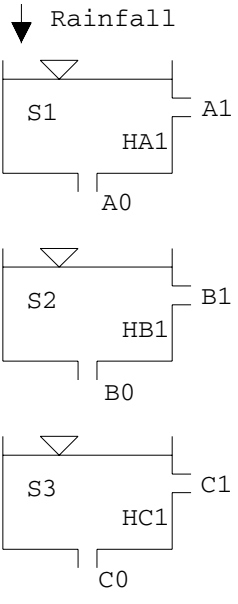


Fig. 1. Tank model diagram showing the three tanks arranged vertically and their water outlets

This model represents the infiltration as a continuous process whenever the water levels (S1, S2, and S3) are greater than zero. At a rainfall event the surface runoff will start once a minimum wetness level is reached (water levels the height of runoff orifices).

The variation in wetness is represented by the volume balance equations which can be expressed as

$$\frac{dS1}{dt} = r - f_1 - Q_1 \tag{1}$$

$$\frac{dS2}{dt} = f_1 - Q_2 - f_2 \tag{2}$$

$$\frac{dS3}{dt} = f_2 - Q_3 - f_3 \tag{3}$$

where r = rainfall (mm/hour). Q1, Q2, and Q3 represent the components of surface runoff from the first, second and third thank respectively. Their amount is linearly proportional to the head of water when it is above the heights of the runoff orifices and can be expressed as

$$Q_1 = A1 \cdot (S1 - HA1) \tag{4}$$

$$Q_2 = B1 \cdot (S2 - HB1) \tag{5}$$

$$Q_3 = C1 \cdot (S3 - HC1) \tag{6}$$

The infiltration rates are, for tank 1, f1, for tank 2, f2, and for tank 3, f3. They are linearly proportional to the head of water and can be expressed as

$$f_1 = A0 \cdot S1 \tag{7}$$

$$f_2 = B0 \cdot S2 \tag{8}$$

$$f_3 = C0 \cdot S3 \tag{9}$$

Although one single tank represents a linear system, the combination of the three tanks results in a nonlinear system. It was shown that, in spite of its simple computational framework, the model is able to reflect the phenomena of rainfall-runoff processes [9].

The NERC HYREX project generated rainfall estimates from a network of raingauges installed across the Brue catchment, in Somerset, South-West England. The river flow was measured at Lovington.

2.2 Artificial Neural Network

The model used in this work is a feedforward ANN, trained by the backpropagation algorithm, which convergence properties are clearly presented in [10]. The selection of the best suited architecture was performed by a Genetic Algorithm (GA) [11] with the parameters shown in Table 1. The ANN architecture was constrained to only three layers and the last layer with only one neuron.

Table 1. GA parameters used for the search of the optimum ANN architecture

Parameter	Range
MUTAT_PROB	0.08
CROSS_PROB	0.8
SELF_ENTERED	0
POP_SIZE	40
ELITISM	1
MAX_GENERATION	50
Neurons in first and second layer	4-26

The search of the optimum ANN architecture was performed by using, as a 'fitness function', the efficiencies sum from the six ANNs applied to simulate the time series from the tank model parameters described in section 2.2. Table 2 describes the parameter values obtained by the GA for the optimum ANN used in this work. The six ANNs were optimized with the same architecture.

Table 2. ANN parameters for the tank model

Parameter	Range
Number of layers Nl	3
Number of neurons on each layer Nn	20 in first layer 10 in hidden layer 1 in last layer
Learning rate η	0.2

2.3 Proposed Model Structure

The objective was to couple one ANN with each one of the nine parameters: A0, BO, CO, A1, B1, C1, HA1, HB1, and HC1. A simple steepest descent algorithm was developed to find the variation of these parameters in order to match the tank model output with the real flow values. Figure 2 shows the time varying data series of HC1 obtained by this method. The variation range for each one of the parameters values was around the fixed value presented in Table 3. It was found that the first three parameters, A0, B0, and C0 had negligible variations, hence, only the other six parameters outputs were used to train the ANNs. The inputs for all the ANNs were rainfall data.

The performance criterion for evaluating the model was the efficiency R2 which is described by equations 10, 11 and 12.

$$R^2 = \frac{F_0 - F}{F_0} \tag{10}$$

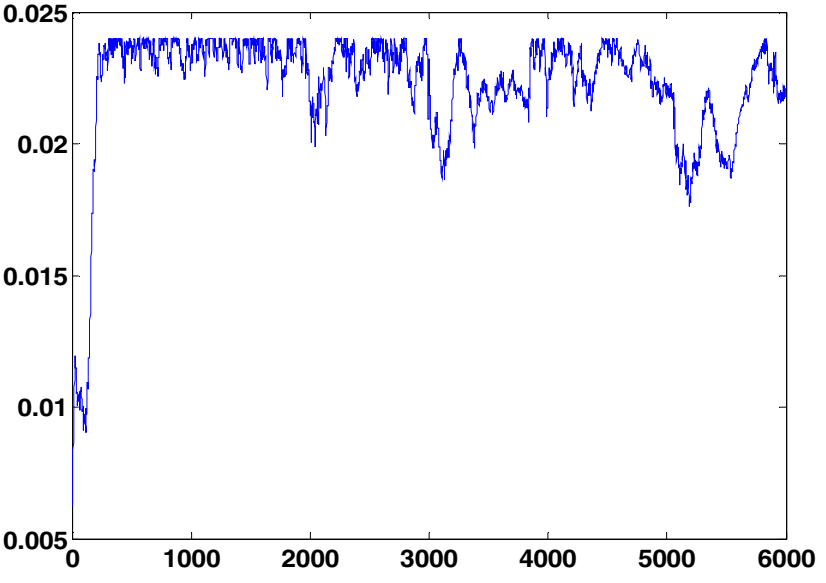


Fig. 2. Flow graphs for real and simulated values. The simulated values were obtained using the fixed parameters indicated in Table 2.

where

$$F_0 = \sum_{k=1}^{K} \left[Q_k - \overline{Q}\right]^2 \tag{11}$$

$$F = \sum_{k=1}^{K} \left[Q_k - \hat{Q}_k\right]^2 \tag{12}$$

here Q, $\hat{Q}$, and $\overline{Q}$ are the observed, computed and mean value of the runoff for the same period. For the graph in Figure 2, the efficiency value was 0.7965.

The second step was to find ideal varying values for each parameter to match the tank model output to the real flow. It was done by using a steepest descent algorithm [12] and the data series for each parameter was used to train the ANNs. It was observed that the parameters A0, B0 and C0 had negligible variations; they were left with a fixed value.

All the algorithms and simulations were developed in Matlab.

3 Results

From the tank model description in section 2.1, the initial set of parameters to work with were A0, B0, C0, A1, B1, C1, HA1, HB1 y HC1. The first step was to search for optimum fixed values for each one of these parameters and the results are presented in Table 3.

Table 3. Tank model fixed parameters

Parameter	Value
A0	0.02095
B0	0.00434
C0	0.000489
A1	0.0000035
B1	0.01575
C1	0.002055
HA1	8.86
HB1	7.2
HC1	0.006

The resulting graph for these values is presented in Figure 3 where it is compared to the real flow data. It should be pointed out that the water levels S1, S2, and S3 were initially set at arbitrary values which affected around the first 1500 hours of the simulation.

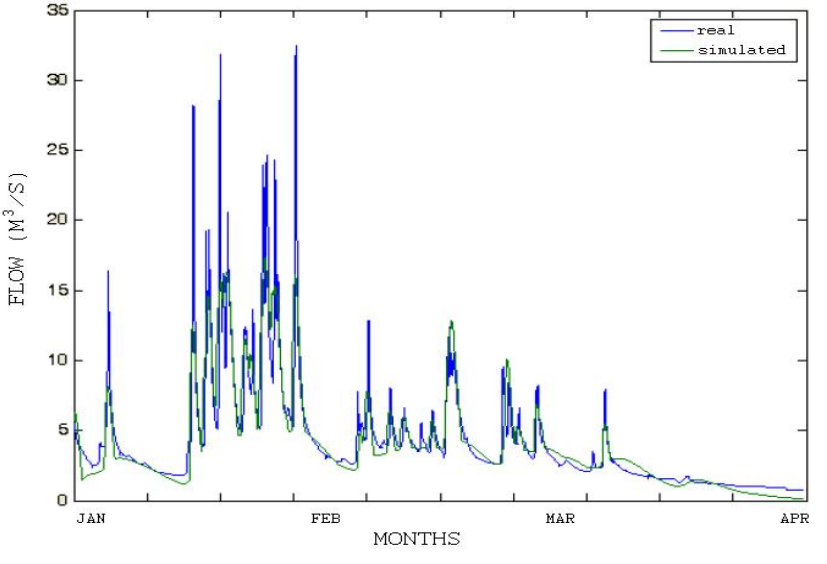


Fig. 3. Flow graphs for real and simulated values. The simulated values were obtained using the fixed parameters indicated in Table 2.

When the six parameters described in section 2.3 were coupled with the ANNs outputs, once the ANNs were trained and tested, the efficiency improved to 0.80143. To train, validate and test the model three different test were used and are described in Table 3.

Table 3. Rainfall and runoff data series time range used to simulated the graphs presented in Figure 2

Set	Time (hr)	Start time	End time
Training	575	19:15 15 Nov 96	18:00 09 Dec 96
Validation	475	14:00 11 Nov 97	08:45 01 Dec 97
Test	475	09:00 01 Dec 97	03:45 21 Dec 97

The resulting graph from the test period is display in Fig. 4.

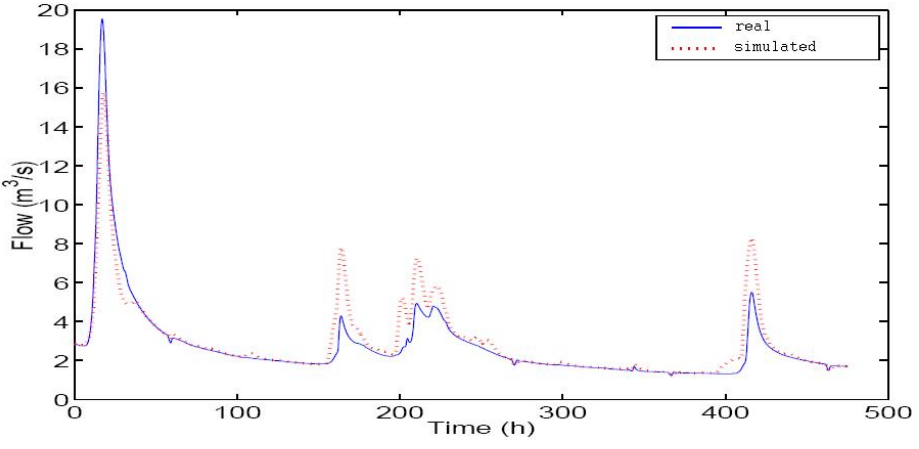


Fig. 4. Flow graphs for real and simulated values. The simulated values were obtained using the variable parameters generated by the ANN.

4 Conclusions

The increased performance efficiency could seem barely noticeable, but it reflects how the ANNs are able to extract information from data series that are affected by different conditions like temperature, soil type, vegetation, human structures, sunlight amount, geology, etc. These conditions are considered in several complex models [13] which can obtain better performance but with costs making them prohibited in many situations. The next step in this work is to couple the ANNs directly to the tank model in order to be trained by the difference between the ANNs outputs and the tank model parameters.

Acknowledgments

The writers wish to express their sincere thanks to all the people who participated and collected the raingauge, flow gauge, and radar data during the HYREX project. This research work has been possible thanks to Subsecretaria de Educación Superior of SEP, who provided the funding through the “Apoyo a la Incorporación de Nuevos PTC del PROMEP”.

References

1. Sugawara, M.: Automatic calibration of the tank model. Hydro. Sci. Bull. 24, 375–388 (1979)
2. Lee, Y.H., Singh, V.P.: Tank Model Using Kalman Filter. Journal of Hydrologic Engineering 4(4), 344–349 (1999)
3. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. of Math. Biophysics, 577–606 (1943)
4. Haykin, S.: Neural Networks – A comprehensive foundation. Prentice-Hall, Inc., Englewood Cliffs (1998)
5. Fine, T.L.: Feedforward neural network methodology (statistics for engineering and information science). Springer, Heidelberg (1999)
6. Cichocki, A., Unbehauen, R.: Neural Networks for Optimization and Signal Processing. John Wiley & Sons, Chichester (1993)
7. Govindaraju, R.S.: Artificial Neural Networks in hydrology: I: preliminary concepts. Journal of Hydrology Engineering 5(2), 115–123 (2000)
8. Govindaraju, R.S.: Artificial Neural Networks in hydrology: II: hydrological applications. Journal of Hydrology Engineering 5(2), 124–137 (2000)
9. Sugawara, M., Watanabe, I., Ozaki, E., Katsuyama, Y.: Tank model with snow component. Research Notes, No. 65, National Research Center for Disaster Prevention, Japan (1984)
10. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-dynamic programming. Optimization and Neural Computation Series, vol. 3. Athena Scientific (1996)
11. Passino, K.M.: Home Page, http://www.ece.osu.edu/~passino/ICbook/ic_code.html
12. Burden, R.L., Faires, J.D.: Numerical Analysis, 7th edn., pp. 628–634 (2001)
13. Mellor, D., Sheffield, J., O'Connell, P.E., Metcalfe, A.V.: A stochastic space-time rainfall forecasting system for real time flow forecasting II: Application of SHETRAN and ARNO rainfall runoff models to the Brue catchment. Hydrology and Earth System Sciences 4(4), 617–626 (2000)

Feature Selection Using Artificial Neural Networks

Sergio Ledesma, Gustavo Cerda, Gabriel Aviña, Donato Hernández, and Miguel Torres

Dept. of Electrical & Computer Engineering. University of Guanajuato. Salamanca, Gto. 36700, México
{selo,gcerdav,avina,donato,mtorres}@salamanca.ugto.mx

Abstract. Machine learning is useful for building robust learning models, and it is based on a set of features that identify a state of an object. Unfortunately, some data sets may contain a large number of features making, in some cases, the learning process time consuming and the generalization capability of machine learning poor. To make a data set easy to learn and understand, it is typically recommended to remove the most irrelevant features from the set. However, choosing what data should be kept or eliminated may be performed by complex selection algorithms, and optimal feature selection may require an exhaustive search of all possible subsets of features which is computationally expensive. This paper proposes a simple method to perform feature selection using artificial neural networks. It is shown experimentally that genetic algorithms in combination with artificial neural networks can easily be used to extract those features that are required to produce a desired result. Experimental results show that very few hidden neurons are required for feature selection as artificial neural networks are only used to assess the quality of an individual, which is a chosen subset of features.

1 Introduction

Feature selection is commonly associated with the curse of dimensionality that affects machine learning and optimization problems solved by backwards induction. Feature selection is important because it may reduce the number of features of a data set, and hence, speed up the learning process while making the data set easier to understand and manage, see [7]. Feature selection may be performed by trial and error of all possible subsets of the original data set, this, of course, is very inefficient and there are several systematic methods to handle the selection process, see [2]. We proposed the use of artificial neural networks (ANN) in combination with genetic algorithms (GA) to assist the search process and select from the data set those most relevant features.

This paper is organized as follows. In Section 2, background information about genetic algorithms and ANN training is reviewed. In Section 3, a simple method to perform feature selection is presented. In Section 4, simulation experiments are performed to verify our results. Finally, Section 5 presents some conclusions and direction for future work.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 351–359, 2008.

2 Background Information

Generally, inductive learning methods work best when supplied with features of a state that are relevant to its evaluation, rather than with just raw state description [10]. Hence, feature selection plays an important role when extracting information from a data set. Unfortunately, features may be correlated and have to be analyzed by groups rather that individually, making the process of feature selection complicated. As an result, in [8], a theory of mutual information that demonstrates the relationship of four selection schemes: maximum dependency, mRMR, maximum relevance, and minimal redundancy is presented; a specific application of this approach is offered in [3].

2.1 The Genetic Algorithm

The genetic algorithm (GA) is a general optimization method that has been applied to many problems including neural network training [9]. Additionally, it has been pointed out in [5] that because the usefulness of the genetic algorithm depends most highly on the representation of the solution, any number of numerical and symbolic problems may be optimized. As it will be shown later, feature selection problems may be solved using genetic algorithms by analyzing the features in groups and reducing highly the number of subsets to evaluate.

2.2 Artificial Neural Networks

They are made out of neurons and are excellent to extract information from a data set even under noisy conditions. During a process, called training, the network's weights are adjusted until the actual network's output and a desired output are as close as possible, see [10]. In general, artificial neural networks can be used to map an input to a desired output, classify data or learn patterns. Hence, artificial neural networks can also be used to assess the quality of an individual (from a genetic algorithm pool), and therefore, perform indirectly feature selection.

3 Proposed Method

There are several existing methods to perform feature selection, see the references in [8]. These methods are based on several criteria: maximum relevance, minimum redundancy, etc. We propose the use of the genetic algorithm and the mean-squared error obtained during the training of a neural network to perform feature selection.

This paper focuses on two typical feature selection problems: mapping and classification. In a mapping problem, a set of input values must produce ("map") a set of desired output values. The feature selection problem consists on reducing the number of variables in the input set while producing the same output. Those values that can be removed from the input set may not contain useful information

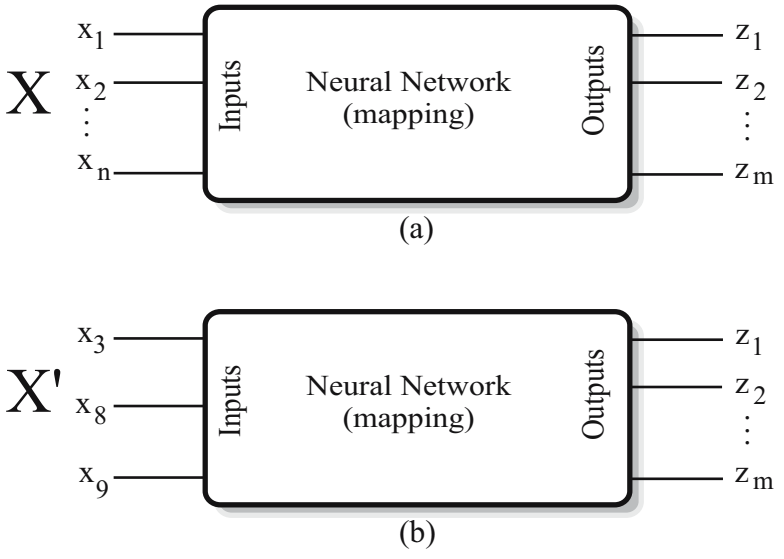


Fig. 1. An artificial neural network used for mapping: (a) using all features, (b) using a subset of features

to produce the desired output or may contain redundant information that is already contained in other input variable (feature) or other set of inputs. This can be observed at Figure 1.a that shows a neural network that maps the input set $X = \{x_1, x_2, \cdots, x_n\}$ into the output set $Z = \{z_1, z_2, \cdots, z_m\}$.

On a classification problem, an input set (feature set) $X = \{x_1, x_2, \cdots, x_n\}$, allows identifying the class that these input values belong. Figure 2.a shows an artificial neural network used as a classifier; the input set $X = \{x_1, x_2, \cdots, x_n\}$ let the neural network identify the class of each element of the input set. In this case, a feature selection problem consists on computing a new input set that is a subset of X, so that only those features that presumably contain useful information to identify the class are considered.

3.1 Coding Feature Selection Using a Genetic Algorithm

Genetic algorithms are non-deterministic algorithms that their success depends strongly on how the problem is coded or mapped into the domain of GA. A feature selection problem may map naturally to a set of binary numbers as it will be explained next.

Consider the input set $X = \{x_1, x_2, \cdots, x_n\}$ with n features, and suppose that there is a subset X' with k features taken from the set X, i.e., $X' = \{x_3, x_8, x_9\}$. Thus, X', a GA individual, can be coded as $0010000110\cdots$, where the first zero indicates that x_1 is not included in X', the third 1 indicates that x_3 is included in X', and so on. That is, those features that included in the subset X' are represented by one, and those features that left out are represented by zero. This simple coding

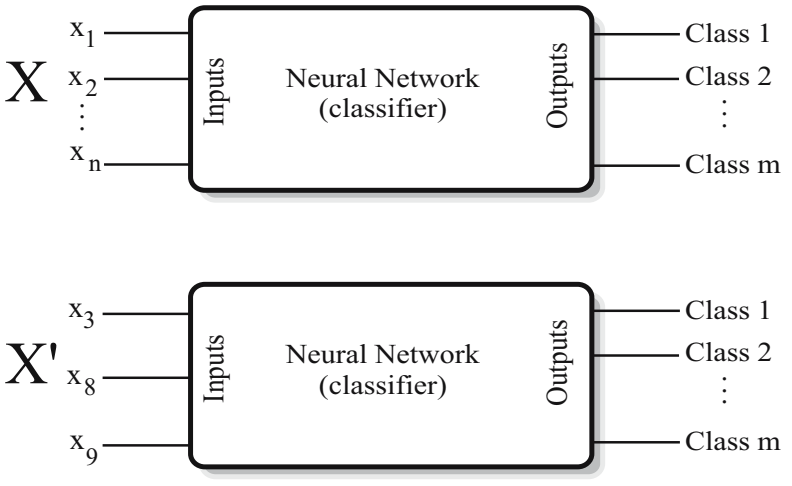


Fig. 2. Artificial neural network used for classification: (a) using all features, (b) using a subset of features

scheme may be used as long as the number of elements of X' be equal to the number of feature to select (k). Finally, it is important to note that each subset of X, a subset of features, may represent an individual with a specific fitness level.

3.2 Initial Pool

A genetic algorithm starts by creating an initial pool of individuals. Some implementations of GA require the creation of more individuals than the size of the initial pool and select only those more fitted individuals. However, when using neural networks for feature selection, it is not appropriate to create more individuals than required as assessing the fitness for each individual is time consuming. More about this will be addressed on subsection 3.4.

To create the initial pool, it is necessary to create a set of individuals; this is a two step process. First, an individual is created by randomly setting k of its bits to one. Second, we check if this individual is already in the pool; if it is, then a new individual is created; if it is not, then this individual is added to the pool. This process continues until the initial pool is full with different individuals.

Once the initial pool is filled, each individual must be evaluated to assess its fitness. Those most fitted individuals will be the parents of the next generation. This will be discussed at the end of this section.

3.3 Reproduction and Mutation

From one generation to the next, the success of GA depends highly on how the individuals are reproduced and mutated. For some implementations of GA,

Table 1. Reproduction used for feature selection

Parent-1 bit	Parent-2 bit	Child bit
0	0	0
0	1	1 with a probability of 0.1
1	0	1 with a probability of 0.1
1	1	1

reproduction and mutation are two separated steps; first the individuals are combined using the probability of crossover, then the new individuals are mutated using the probability of mutation. On the other hand, when using GA for feature selection, these two operations must be performed together as the number of features from generation to generation must remain constant.

Once the individuals from the current generation are evaluated, a list of possible parents is created. This list includes all individuals of the current generation, and a fitness level associated to each individual. Combining the probability of crossover and the fitness level of each individual two parents are chosen. Every time an individual is selected as a parent, its probability of selecting him again is reduced. After this, two new children are produced from these two parents. The process continues until the new generation is completely created.

In order to produce the children, the following algorithm was used. Suppose there are two parents X_1' and X_2', thus a bit position is randomly generated and the children bit is set using the rules on Table 1. Basically, if both parents include that feature, the child's bit is set to one; if non of the parents include that feature, this bit is set to zero; and if only one of the parents include this feature, this bit is set to one with a probability of 0.1. Right after this, the gene mutation occurs: a bit position is randomly generated and the child bit on this position is mutated with a given probability. This process continues until the child contains k bits set to one (k features). Once the first child has been created, a second child is created using the same mechanism just described.

After carefully observing Table 1, it can be seem odd to set a bit to 1 with a probability of 0.1 when only one of the parents has that feature. GA is based on diversity and fitness; a careful balance must exist between these two. If this probability is set to 0.5 or more the algorithm may not favor features that are present on both parents. If this probability is set to a value close to 0, some good features will die too soon killing diversity.

3.4 Fitness of an Individual

In order to choose the parents for the new generation and improve the new pool of individuals, it is very important to assign a fitness level to each individual. This process can be tricky as some criteria must be used.

Consider again Figure 1.a, in this case the artificial neural network is trained by applying it a specific input, and observing the actual network output. The training process uses the mean squared error (mse) measured between the actual

network and the desired output. The *mse* is typically computed using all ANN outputs and all training cases. Consequently, the learning process is managed by the value of the *mse*. When the *mse* reaches a desired value, the neural network has been trained and the ANN can be used to solve real life problems.

Figure 1.a shows an artificial neural network that is being trained using all features from the data set. Figure 1.b shows an artificial neural network that is being trained using only a subset of features.

Because of their fancy cleverness to discover hidden patterns and structures (even under noisy conditions), we propose the use of artificial neural networks to evaluate the fitness of an individual. That is, a given subset of features is used for training an artificial neural network; if the subset of features contains features that help the network to map the desired output, the resulting *mse* will be small. On the other hand, if this subset does not have information to get the desired output, the resulting ANN *mse* will be large.

Once the ANN learning process has completed, the individual's *mse* must be converted so that a very small *mse* maps to a high fitness level; the exponential function was used for this purpose see [6]. Finally, a list of parents for the next generation must be created using each individual's fitness. The parents with high fitness levels will have a high expected frequency of being selected from this list.

Consider now Figure 2 that shows an artificial neural network used for classification. Figure 2.a shows an artificial neural network that is being trained for classification using all the features in the data set; while Figure 2.b shows an artificial neural network that is being trained for classification using only a subset of features. For this case, the *mse* can be calculated using the fact that only one output neuron must be active. If the ANN activation function is the $logsig(x)$, the active value should be approximately 0.9 and the inactive value 0.1. On the other hand, if the ANN activation function is the $tanh(x)$, the active value should be approximately 0.9 and the inactive value -0.9.

4 Simulation Results

On machine learning, some practitioners recommend splitting the available data into two sets: one for training and another for validation. However, in some real life problems, there are awfully few data available to perform training and validation. Besides, it is possible that these two sets are very different or extremely small; making difficult to reach any valid conclusion from the data analysis. One typical solution, for short data set analysis, is to use the leave-one-out cross validation (LOOCV), which is a $K - fold$ cross validation with K being equal to the number of observations in the data set. Leave-one-out cross-validation consists on using only a single observation from the original data set for validation, and leaving the remaining observations for training. This process goes over until each observation in the data set is used once for validation.

LOOCV is useful for classification problems and has been used in some previous papers to test feature selection. Unfortunately the sets, used there, have relatively very few training cases are not useful to train artificial neural networks

without easily incurring in over-fitting. Consider for example the lung data set used in [3] with 325 features, 7 classes and only 73 training cases; even the simplest neural network (with very few hidden neurons) would commit overfitting. Similarly, the data sets of [1] or [4] are not most favorable for ANN learning. Therefore, some synthetic data sets are created and tested to validate our results.

4.1 Data Set with Uncorrelated Features

To evaluate the usefulness of the proposed method, we carried out experiments on a synthetic data set for a mapping problem with 60 input features, 5 outputs and 200 training cases. The input values, in this case the features, where a set of random values $X = \{x_1, x_2, \cdots, x_n\}$ uniformly distributed in $[-\pi, \pi]$ while the output values were synthesize as shown in table 2. Note that the desired output of the mapping problem was computed using only: x_{10}, x_{20}, x_{30}, x_{40}, and x_{50}; while the other input features were not used at all to compute the output. In other words, the problem consisted on selecting only 5 features from the 60 contained in the original data set while producing the desired output.

Once the data set was synthesize, a multi-layer feed-forward artificial neural network with five inputs, five outputs, and 9 hidden neurons was created and trained following the genetic algorithm to perform feature selection by the proposed method. During training, the ANN required 500 epochs or less to reach an mse of 0.0001. Because of the network size, finding the fitness of an individual may take some seconds when running on a computer Pentium 4 at 3.2 GHz.

It is important to mention that the number of hidden neurons is critical. If the number of hidden neurons is small, the resulting *mse* will be the same no matter what features are applied to the input. Therefore, the number of hidden neurons should be kept to the maximum just before committing over-fitting. This means that the number of training cases directly affects the number of hidden neurons. The software *NeuralLab* that is available at *Wikipedia* under "neural network software" provides the appropriate number of hidden neurons to avoid over-fitting. It is also possible to estimate the maximum number of hidden neurons once the number of training cases is given, because in order to compute the network weights (or unknowns) a minimum number of equations (training cases) is required.

The initial population was set to 100 individuals, and once the algorithm started, it was after only 8 generations by the time the algorithm selected: x_{10}, x_{20}, x_{30}, x_{40}, and x_{50} (the only features that were used to compute the output).

Table 2. Synthetic data set with uncorrelated features

Output	Value
1	$\sin(x_{10})$
2	$\sin(x_{20})$
3	$\cos(x_{30})$
4	$\cos(x_{40})$
5	$\cos(x_{50})$

Finally, note that the proposed method allowed a reduction from $5,461,512$ (as computed in Equation 1) to only 800 evaluations to select those features required to get the desired output.

$$\binom{60}{5} = 5,461,512 \tag{1}$$

4.2 Data Set with Correlated Features

To make the problem more difficult, we repeat the same experiments of subsection 4.1 but using correlated outputs as shown in Table 3. For this case, one of the outputs was computed using two inputs making more difficult to solve this feature selection problem.

Table 3. Synthetic data set with independent features

Output	Value
1	$\sin(x_{10})$
2	$\sin(x_{20})$
3	$\cos(x_{30} + x_{40})$
4	$\cos(x_{40})$
5	$\cos(x_{50})$

Once the data set was synthesize, an artificial neural network with five inputs, five outputs, and 9 hidden neurons was created and trained just as before to perform feature selection. The initial population was set to 100 individuals. However and because of the added correlation in the inputs, it took 10 generations for the proposed algorithm to select: x_{10}, x_{20}, x_{30}, x_{40}, and x_{50}.

5 Summary

Feature selection simplifies some mapping and classification problems by reducing the number of features to analyze. The proposed algorithm does not focus on the relevance or redundancy of the features; instead it passes this job to an artificial neural network, because they have excellent skills to discover hidden patterns yet under noisy conditions.

We have shown that a genetic algorithm can be used to assist the search for those feature that will produce the desired results. A GA individual is a subset of features built from the main set of features. Each individual may be evaluated using the mean squared error obtained during the training of an artificial neural network. The mean squared error can be, then, used to select those more fitted individuals to create a new generation of individuals. The experimental results show that the proposed method allows extracting those features that are required to get a desired output. The experimental results were performed using uncorrelated and correlated features. Last, note that artificial neural network

training is time consuming, and it would interesting to incorporate other artificial intelligence techniques to reduce the total processing time of this algorithm.

References

1. Alizadeh, A.A.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403, 503–511 (2000)
2. Bonev, B., Escolano, F., Carzola, M.A.: A Novel Information Theory Method for Filter Feature Selection. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 431–440. Springer, Heidelberg (2007)
3. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology 3(2), 185–205 (2005)
4. Garber, M.E., Troyanskaya, O.G., et al.: Diversity of gene expression in adenocarcinoma of the lung. PNAS USA 98(24), 13784–13789 (2001)
5. Jones, M.T.: AI Application Programming, Charles River Media, 2nd edn., pp. 229–261 (2005)
6. Masters, T.: Practical Neural Network Recipes in C++, pp. 135–164. Academic Press, Inc., London (1993)
7. Peng, H., Ding, C., Long, F.: Minimum redundancy maximum relevance feature selection. IEEE Intelligent Systems 20(6), 70–71 (2005)
8. 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), 1226–1238 (2005)
9. Reed, R.D., Marks II, R.J.: Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks, pp. 185–195. The MIT Press, Cambridge (1999)
10. Russel, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, pp. 109–134. Prentice-Hall of India, Englewood Cliffs (2006)

Composite Kernels for Support Vector Classification of Hyper-Spectral Data

M. Kohram and M.N.M. Sap

Faculty of Computer Science and Information Systems, University Technology Malaysia
kmojtaba2@siswa.utm.my,
mohdnoor@utm.my

Abstract. The incorporation of prior knowledge into the Support Vector Machine (SVM) architecture is a problem which if solved can lead to much more accurate classifiers in the near future. This result could be particularly effective in the classification of remote sensing imagery, where an abundance of information is available prior to classification. The most evident method to feed prior knowledge into the SVM algorithm is through the SVM kernel function. This paper proposes several composite kernel functions designed specifically for land cover classification of remote sensing imagery. These kernels make use of the spectral signature information, inherently available in remote sensing imagery. The results achieved from these kernels are very much satisfactory and surpass all previous results produced by classical kernels.

1 Introduction

Support vector machines [1] have been successfully applied to multi-spectral and hyper-spectral data classification [2], [3], [4]. Their ability to map data into very high dimensions without much added computational complexity and without falling victim to the Hughes phenomenon (*curse of dimensionality)* have made them an ideal method for dealing with remote sensing data. In addition, the solution sparseness (only part of the database is used for classification) and few number of tuning parameters available have made this technique a very attractive classification method for researchers.

A setback of the SVM algorithm is that there are few techniques available for incorporating prior information into its formulation. In the case of remote sensing data this is very unfortunate due to the vast amount of prior information available in remote sensing imagery. One method for incorporating prior information into SVMs is the usage of appropriate similarity measures in the SVM kernel. The typical SVM formulation utilizes either the Euclidean distance or the dot product of the data vectors for discrimination purposes. While these measures have proven effective in many cases, when dealing with remote sensing data they do not take into account the spectral signatures of different land cover classes.

In this paper, two measures of similarity, namely the Spectral Angle Mapper (SAM) and the Spectral Information Divergence (SID) are introduced. These measures have been devised specifically to make use of the vast amount of information available in hyper-spectral imagery. These measures are then embedded into the SVM architecture and subsequently mixed together to achieve new discriminant measures which achieve

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 360–370, 2008.

high separability when applied to hyper-spectral imagery. The idea of merging these kernels comes from the fact that each of these kernels performs in a different manner when applied to a data set, meaning that accuracy results on the various classes is very dissimilar and hence a mixture of these kernels might result in better generalization.

In the next section a brief introduction to SVMs is provided. In section 3 the "kernel trick" and nonlinearity in SVMs is introduced. Section 4 takes a deeper look at the qualifications of the two similarity measures mentioned above. Subsequently, section 5 gives a brief overview of kernel mixtures and introduces the composite kernels proposed in this paper. Section 6 provides experimentation and results followed by a brief discussion and conclusion in section 7.

2 Support Vector Machines

This section is dedicated to a brief introduction of SVMs and their classification process. We will start with basic binary and linear classification and move on to the methods adopted for removing these restrictions. For further discussion and a deeper understanding, readers are referred to [1], [5].

Assume that the training set consists of N data points $\mathbf{x}_i$ where $\mathbf{x}_i \in \Re^d$, (i = 1, 2, …, N). To each vector we assign a target $y_i \in \{-1,1\}$. Our task is to locate a hyper-plane $\mathbf{w} \cdot \mathbf{x}^T + b$ which correctly divides these data points into two classes, resulting in the decision function $f(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x}^T + b)$ which in turn correctly classifies the data. In effect, there exist an infinite number of hyper-planes which correctly divide the data points. Among these, SVMs select the hyper-plane which *maximizes the margin*. The margin is defined as the distance between the classifier and the closest training points. The *maximal margin hyper-plane* is identified by solving the following convex optimization problem with respect to w, ξ, and b:

$$\begin{aligned} &\textit{Minimize}: \\ &\quad \|\mathbf{w}\| + C\sum_{i=1}^{N}\xi_i \\ &\textit{Subject to}: \\ &\quad y_i(\mathbf{w}\cdot\mathbf{x}_i^T + b) \geq 1 - \xi_i \\ &\quad \xi_i \geq 0 \\ &\quad i = 1,\ldots,N \end{aligned} \tag{1}$$

The slack variables ξ_i are added to the formulation to allow for misclassifications when the two classes are inseparable. The above formulation maximizes the margin $(1/\|\mathbf{w}\|)$ while minimizing the errors made during classification by placing a penalty term, C, on the summation of errors made by the classifier. This optimization problem is solved by transforming it to its corresponding dual problem. Following this adaptation, problem (1) is converted to the following optimization problem:

$$\begin{aligned} &\textit{Minimize}: \\ &\quad W(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j^T) \\ &\textit{Subject to}: \\ &\quad \sum_{i=1}^{N} y_i \alpha_i = 0 \\ &\quad 0 \leq \alpha_i \leq C, i = 1,\ldots,N \end{aligned} \tag{2}$$

whereby α_i represent the lagrange multipliers and are bounded by the penalty parameter C, hence resulting in the constraints occasionally being called the *box constraints*. This formulization results in a decision function of the form:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}^T) + b^*\right) \tag{3}$$

It can be seen that only data vectors with $\alpha_i \neq 0$ play a role in finding the optimal hyper-plane. Usually most of the α_i in the optimal solution are equal to 0, and this results in a huge amount of data not being responsible for classification. Only a select group of vectors called the support vectors contribute to the solution. This feature (sparseness) of SVMs is very valuable when dealing with large datasets like hyperspectral and multispectral images.

While the above formulation is designed for binary classification, various techniques have been devised to expand this to multi-class classification. The two most popular methods are the *one against all* and the *one against one*. The one against all technique forms a binary SVM for each class against all the other classes while the one against one method forms a binary SVM for each pair of classes. Subsequently for each testing point the most often computed label is assigned to the particular vector.

3 Kernels and Nonlinear Machines

The problem dealt with up to now has been to linearly classify the data, but for nonlinear classifications SVMs make use of kernel methods. These methods consist of mapping data into a high dimensional feature space $\Phi(\mathbf{x})$ where the data can be smoothly classified by a linear machine.

Taking note of equations (2) and (3), it is observed that wherever the data points appear they are in form of dot products, thus it is convenient to define a function K as:

$$K(\mathbf{x}, \mathbf{x}_i) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i) \tag{4}$$

This function is called the kernel function and results in the dot product of the training vectors in feature space. In this manner, instead of taking on the computationally expensive task of mapping all the data points into feature space, the kernel matrix can be computed with much more efficiency and speed. This results in an efficient nonlinear classification of the data. By replacing the data points with the kernel function in (2), the following formulation is achieved:

$$W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j K(\mathbf{x}, \mathbf{x}_i) \tag{5}$$

while the decision function can be expressed as:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i^* y_i \, K(\mathbf{x}_i, \mathbf{x}_j) + b^*\right) \tag{6}$$

For a function to be categorized as a kernel that function has to fulfill Mercer's conditions [1]. These conditions are equivalent to requiring that "for any finite subset of input space $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, the corresponding matrix is positive semi-definite" [5].

4 Spectral Similarity Measures

While classical kernels have proven successful in most cases, when dealing with hyper-spectral data, they do not take full advantage of the rich amount of *a priori* information available. This could be due to the fact that these Kernels do not take into account the band to band spectral signature effects. Depending on their localism or globalism, classical kernels mostly use the either the Euclidean distance (local) or the dot product (global) of two vectors as their similarity measure. In order to effectively make use of information intrinsically available in remote sensing imagery, measures of similarity other than the Euclidean distance and dot product must be used. Here we introduce two measures that are specifically designed for this purpose. It is insightful to add that while one of these measures is geometrical and another is entropy-based, both these measures are insensitive to the length of the pixel vectors whose similarity they measure.

4.1 The Spectral Angle Mapper (SAM)

SAM has been abundantly used for remote sensing applications. It measures the similarity of two vectors by finding the angle between two data vectors. The SAM function is defined as:

$$\alpha(\mathbf{x},\mathbf{x}_i) = arc\cos\left(\frac{\mathbf{x}\cdot\mathbf{x}_i^T}{\|\mathbf{x}\|\|\mathbf{x}_i\|}\right) \tag{7}$$

SAM has the ability to exploit spectral characteristics while being robust to energy differences because the angle between two vectors is not affected by the length of the particular vector [6]. However, if the angle between two data vectors is very small, SAM tends towards the Euclidean distance and the two measures achieve very similar discrimination results [7].

4.2 Spectral Information Divergence (SID)

Spectral information divergence [8] is essentially an application of the *Kullback-Lieber* (KL) divergence (Kullback-Lieber distance, relative entropy) to hyper-spectral data. The KL divergence is an information theoretic based measure that is defined as the "distance" of one probability distribution from another. It is part of a broader class of divergences called f-divergences [9] utilized to quantify "closeness" of one probability distribution from another. For two discrete probability distributions P and Q, the f-divergence can be written as:

$$I_f\left(P,Q\right) = \sum_i q(x_i) f\left(\frac{p(x_i)}{q(x_i)}\right) \tag{8}$$

By substituting $f\left(t\right) = t\ln t$ into the above, the KL divergence is achieved:

$$D(P\,\|\,Q) = \sum_i p(x_i)\log\left(\frac{p(x_i)}{q(x_i)}\right) \tag{9}$$

It can be proven that the inequality $D(P\,\|\,Q) \geq 0$ always holds and that $D(P\,\|\,Q) = 0$ if and only if $P = Q$. It can also be seen that in general the KL divergence

is not symmetric, that is, in general $D(P \| Q) \neq D(Q \| P)$. This can easily be overcome by defining:

$$I(P,Q) = D(P \| Q) + D(Q \| P) \tag{10}$$

as the distance between distributions P and Q. Loosely speaking in information theoretic terms, this equation is equivalent to the amount of *uncertainty* resulting from evaluation of the *entropy* of P by use of distribution Q thus quantifying the distance between P and Q.

To employ the KL divergence as a measure of distance between two data points, a probabilistic interpretation of the data points has to be adopted. This can be achieved by viewing the spectral signature of each pixel vector as a probability distribution and subsequently using the KL divergence to measure the distance of these distributions. Each pixel vector's probability distribution can be derived by normalizing the pixel vector to unity. For a vector $\mathbf{x} = [x_1, ..., x_d]^T$ this is accomplished as follows:

$$p_i = \frac{x_i}{\sum_{i=1}^{d} x_i} \tag{11}$$

The above formulation results in a probability distribution vector $\mathbf{p} = [p_1, ..., p_d]^T$ for each pixel, which from here on describes the properties of the pixel vectors. This adjustment is significant since it opens many probabilistic and statistical routes for processing the data. As an example it can be seen that for each pixel, statistics of different order such as mean, variance, third and fourth order statistics can be defined and utilized in an appropriate manner [7]. Using this metric the spectral information divergence of two pixel vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ with probability vectors $\mathbf{p}$ and $\mathbf{q}$ respectively, is defined as:

$$SID(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_i \| \mathbf{x}_j) + D(\mathbf{x}_j \| \mathbf{x}_i) \tag{12}$$

Where:

$$D(\mathbf{x}_i \| \mathbf{x}_j) = \sum_{l=1}^{d} p_l \log(p_l / q_l) \tag{13}$$

To illustrate how SID takes into account spectral signatures, we can consider the extreme case of the distance between pixel vector $\mathbf{x}$ and its probability distribution vector $\mathbf{p}$. Since both these vectors have the same spectral signature, we are hoping that the distance between them is minimal using every metric. Since SAM is also length insensitive, it does achieve the desired results, but this is not the case if Euclidean distance is applied to these two vectors. Using the Euclidean distance, these two vectors might readily be cast very far from each other.

To sum up, SID does not take into account the geometrical features of the pixel vectors; instead it takes note of the discrepancy between the probability distributions produced by each pixel vector. It is hoped that this unique feature of SID will render this measure more effective than the geometrical measures during classification.

5 Mixture Kernels

By embedding the above similarity measures into an Radial Basis Function (RBF), these measures can be effectively used as the kernel function for the SVM algorithm. The three similarity measures applied in this paper are the Euclidean distance, spectral angle mapper and the spectral information divergence. The insertion of these measures into the RBF function results in the following kernels:

1) Euclidean distance based Gaussian RBF (basic RBF):

$$K(\mathbf{x},\mathbf{x}_i) = \exp(-\gamma \left\| \mathbf{x} - \mathbf{x}_i \right\|^2) \tag{14}$$

2) SAM based RBF [10], [11], [12], [13]:

$$K(\mathbf{x},\mathbf{x}_i) = \exp(-\gamma \left| \alpha(\mathbf{x},\mathbf{x}_i) \right|) \tag{15}$$

3) SID based RBF [10], [14], [15]:

$$K(\mathbf{x},\mathbf{x}_i) = \exp(-\gamma\, SID(\mathbf{x},\mathbf{x}_i)\ ) \tag{16}$$

While the above kernels have provided individually impressive results, it has been shown that each kernel reveals individual properties that are unique to that kernel [15]. In this view, a mixture of these functions which takes advantage of the diversity and unique qualities of each kernel could prove productive. Mixtures or composites can be achieved by simple or weighted addition of the aforementioned kernels. It must be noted that all simple and weighted additions of two or more mercer kernels result in an eligible mercer kernel [16]. Composites of classical functions as in the polynomial and RBF kernels have shown acceptable results in application [17], [18] and hence it is reasonable to perceive that a mixture of functions with higher classification accuracy than the classical kernels will result in an improved and more accurate classifier.

In this paper, using the above kernels, four mixtures are produced and the results of these mixtures are compared from various perspectives. The formed kernels are derived by simple addition of the functions above. Thus for example the SID-SAM mixed kernel would simply be:

$$K(\mathbf{x},\mathbf{x}_i) = \exp(-\gamma_1\, SID(\mathbf{x},\mathbf{x}_i)\ ) + \exp(-\gamma_2 \left| \alpha(\mathbf{x},\mathbf{x}_i) \right|) \tag{17}$$

In this same manner the other three kernels are the SAM-RBF, SID-RBF, and the SAM-SID-RBF kernels. For ease of naming convention, and since the RBF is usually used with the Euclidean distance, the word RBF is used instead of Euclidean in the naming of the functions. Also, note that the fourth function is a threefold mixture of all the above kernels. While the overhead of such a function might not be worth the improvement in accuracy, it could be useful to take a look at the results achieved from this kernel as opposed to the results from the two fold mixtures. Experimentation and results on these kernels are provided in the next section.

6 Experimentation and Results

Experiments were applied to the well known hyper-spectral AVIRIS image taken over Northwest Indiana's Indian Pine in June 1992 [19]. This image consists of 145 × 145 pixels in 220 spectral bands. Figure 1 shows band 100 along with the land cover map of the image. The image consists of sixteen classes of which seven were discarded due to insufficient data. The remaining nine classes were used for classification purposes.

Before any testing could begin, parameter tuning had to be performed on the various kernels to pinpoint exact values for C and γ of each kernel. This was achieved through 10-fold cross validation of a randomly selected training set consisting of 50 data points from each class. A useful observation made in this phase was that simultaneous tuning of the parameters essentially reaches the same results as tuning each kernel separately and using the obtained parameter in the mixture kernel. Utilization of this practical remark leads to a two dimensional grid search at all times instead of dealing with a time consuming three or four dimensional grid search in the parameter tuning phase.

For the first phase of testing, 10 independently selected training and testing sets, each dividing the data into two equal parts, were acquired from the data and the seven kernels were tested on these ten sets. Table 1 depicts classification results for the first five sets while table 2 shows cumulative results for all data sets.

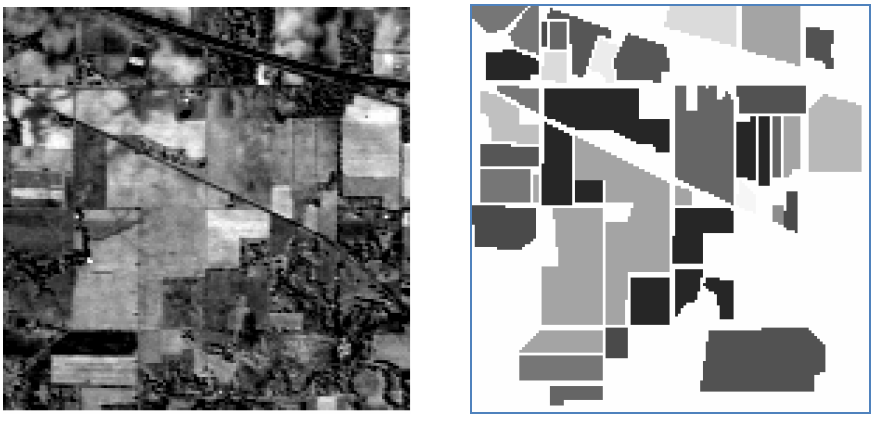


Fig. 1. (*right*) band 100 of the AVIRIS image in grayscale, (*left*) thematic map of the AVIRIS image

The first column of table 1 in each set depicts the accuracy of the specific kernel on that data set and the second column is the number of support vectors used for classification with this kernel. From table 1 and table 2 it can be clearly seen that among the non-mixture kernels the SID kernel outperforms the others by a comparatively large margin. This result is in line with other studies performed on this kernel [20], [15]. However by looking at table 1 it is not clearly evident which of the two-kernel mixtures have higher accuracy. RBF-SAM and RBF-SID are each the best on two data sets while SAM-SID is the best on one of the data sets. A look at table 2 will reveal

that while all three of these kernels achieve close results, the RBF-SID is the more accurate classifier and moreover, this kernel has the minimum number of support vectors among the three kernels. This leads to a very sparse solution which is more time-efficient and also has higher generalization capability. Lastly, by looking at the RBF-SAM-SID kernel, it can be viewed that as expected this kernel outperforms all the previous kernels in terms of accuracy. In table 1 it can be seen that this kernel is first in terms of accuracy on all but two data sets (set 3).

Table 1. Comparative results of the various kernels on the first five data sets. In each set the most accurate classifier is set in bold.

	Data Set									
	1		2		3		4		5	
Kernel	Acc	nSV	Acc	nSV	Acc	nSV	Acc	nSV	Acc	nSV
RBF	94.26	1584	94.58	1578	94.09	1588	94.64	1584	94.77	1631
SAM	94.07	2476	93.98	2482	94.04	2514	94.30	2496	94.58	2546
SID	95.07	1860	94.88	1866	**95.16**	1900	94.82	1882	95.61	1886
RBF-SAM	95.01	1757	**95.35**	1780	94.77	1788	95.16	1765	95.29	1811
RBF-SID	95.33	1717	95.20	1707	94.82	1741	95.07	1740	95.65	1750
SAM-SID	95.11	1921	94.97	1929	95.05	1962	94.78	1934	95.63	1943
RBF-SAM-SID	**95.37**	1764	95.24	1737	94.90	1766	**95.18**	1769	**95.69**	1788

In all 10 sets it came second only three times, outperforming all the other kernels on the other 7 sets. Also in terms of number of support vectors this kernel achieved a respectable second.

Another experiment undertaken was to do a 20-fold cross validation on the whole data set (9345 pixels) to mainly confirm the above results and also to get a deeper understanding of the gap between the kernels. Results of this experiment are presented in the third column of table 2. Sure enough, the RBF-SAM-SID kernel is the strongest in this part but because the volume of the training data in each fold is very large the difference between the classifiers accuracy is small. Here again the SID shows its power by coming very close to the mixture models in terms of accuracy. The SAM-SID and the RBF-SAM show exactly the same result and again the RBF-SID shows stronger than these two kernels.

Table 2. Average number of support vectors and average accuracy over 10 randomly selected training and testing sets. The third column is the result of a 20 fold cross validation experiment. The highest accuracies are emboldened.

Kernel	Avg. nSV.	Avg. Acc.	20-fold CV
RBF	1593	94.42	95.73
SAM	2503	94.20	95.53
SID	1879	94.97	96.47
RBF-SAM	1778	95.09	96.55
RBF-SID	1727	95.25	96.63
SAM-SID	1924	95.02	96.55
RBF-SAM-SID	1763	**95.31**	**96.67**

In the second phase of testing we will look at how these kernels react to variation in the number of training samples provided. For this purpose the data is first divided into 5 sets, with training sets consisting of 5,10,30,50 and 70 percent of the data respectively. The results of this experiment are shown in Figure 2.

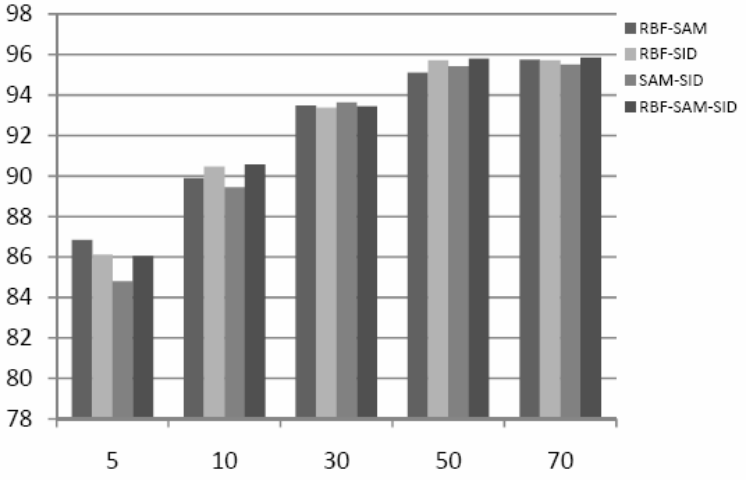


Fig. 2. Effects of variation in training size on the various kernels

Figure 2 illustrates that the trend of classification accuracy does not significantly change with variation in training set size. However, there is a mild change of trend in the 5 percent case. In this case for the first time in experimentation the RBF-SAM-SID kernel comes 3rd in accuracy. While the two kernels occupying first and second place both have a Euclidean distance particle in them. This demonstrates the high ability of the Euclidean based kernel to generalize from very low amounts of training data. After this oddity, the rest of the diagram follows the trend shown before in the basic training and testing phase, meaning that the RBF-SID and RBF-SAM-SID kernels mostly dominate the other kernels.

7 Discussion and Conclusion

In this paper several spectral similarity measures were mixed and composites of these were used as the kernel function in the support vector machine. The results showed an increase in accuracy occurs when more intelligent and complete information about the state of the data is correctly fed into an algorithm. The mixture of the SID measure with the Euclidean distance based measure yielded the highest accuracy among the two-kernel mixtures while as expected; the three kernel mixture yielded the maximum accuracy. Future directions for this research could be toward other similarity measures which exploit the spectral signature of land cover in remote sensing imagery. The definition of a pixel vector as a probability distribution which is provided in section 4.2 opens a very wide field for discovering new similarity measures. Utilizing this

definition, many "distances" and divergences that are being used in probability theory can be readily applied to the problem at hand.

The unique properties of each of these kernels, also brings to mind the idea of an ensemble of classifiers using the various kernels. An ensemble is most effective when the different experts involved are most dissimilar to each other hence achieving maximum generalization capability in this manner. The dissimilarity, yet effectiveness of these kernels can prove a decisive factor in forming ensembles of these kernels.

Acknowledgments

This research was supported by the Research Management Center, University Technology Malaysia (UTM) and the Malaysian Ministry of Science, Technology and Innovation (MOSTI) under vote number 79094.

References

1. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
2. Keuchel, J., Naumann, S., Heiler, M., Siegmund, A.: Automatic land cover analysis for Tenerife by supervised classification using remotely sensed data. Remote Sensing of Environment 86, 530–541 (2003)
3. Shah, C.A., Watanachaturaporn, P., Varshney, P.K., Arora, M.K.: Some recent results on hyperspectral image classification. In: 2003 IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data (2003)
4. Pal, M., Mather, P.M.: Assessment of the effectiveness of support vector machines for hyperspectral data. Future Generation Computer Systems 20, 1215–1225 (2004)
5. Cristianini, N., Shaw-Taylor, J.: Support Vector Machines. Cambridge University Press, Cambridge (2000)
6. Kruse, F.A., Richardson, L.L., Ambrosia, V.G.: Techniques Developed for Geologic Analysis of Hyperspectral Data Applied to Near-Shore Hyperspectral Ocean Data. In: Fourth International Conference on Remote Sensing for Marine and Coastal Environments, Orlando, Florida (1997)
7. Chang, C.-I.: Hyperspectral Imaging: Techniques for Spectral Detection and Classification. Kluwer Academic/Plenum Publishers, New York (2003)
8. Chang, C.-I.: An information theoretic-based measure for spectral similarity and discriminability. IEEE transactions on geoscience and remote sensing 46, 1927–1932 (2000)
9. Ali, S.M., Silvey, S.D.: A General Class of Coefficients of Divergence of One Distribution from Another. Journal of the Royal Statistical Society 28, 31–142 (1966)
10. Mercier, G., Lennon, M.: Support Vector Machines for Hyperspectral Image Classification with Spectral-based kernels. In: IEEE International Geoscience and Remote Sensing Symposium (2003)
11. Fauvel, M., Chanussot, J., Benediktsson, J.A.: Evaluation of Kernels for Multiclass Classification of Hyperspectral Remote Sensing Data. In: IEEE International Conference on Acoustics, Speech and Signal Processing (2006)
12. Sindhumol, S., Wilscy, M.: Hyperspectral Image Analysis -A Robust Algorithm using Support Vectors and Principal Components. In: The International Conference on Computing: Theory and Applications (2007)

13. Sap, M.N.M., Kohram, M.: Spectral Angle Based Kernels for the Classification of Hyperspectral Images Using Support Vector Machines. In: IEEE proceedings of the Asia Modelling Symposium 2008 (2008)
14. Nemmour, H., Chibani, Y.: Multiple support vector machines for land cover change detection: An application for mapping urban extensions. ISPRS Journal of Photogrammetry & Remote Sensing 61, 125–133 (2006)
15. Kohram, M., Sap, M.N.M., Ralescu, A.L.: Spectral Information Divergence based kernels for SVM Classification of Hyper-Spectral Data. In: The 19th Midwest Artificial Intelligence and Cognitive Science Conference, Cincinnati, OH, USA (2008)
16. Scholkopf, B., Smola, A.J.: Learning with kernels. MIT Press, Cambridge (2002)
17. Smits, G.F., Jordaan, E.M.: Improved SVM Regression using Mixtures of Kernels. In: Proceedings of the 2002 International Joint Conference on Neural Networks, Honolulu, HI, USA (2002)
18. Jiang, T., Wang, S., Wei, R.: Support Vector Machine with Composite Kernels for Time Series Prediction. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4493, pp. 350–356. Springer, Heidelberg (2007)
19. Landgrebe, D.: AVIRIS NW Indiana's Pines 1992 dataset (1992), `http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/documentation.html`
20. van der Meer, F.: The effectiveness of spectral similarity measures for the analysis of hyperspectral imagery. International Journal of Applied Earth Observation and Geoinformation 8, 3–17 (2006)

A Constraint-Handling Genetic Algorithm to Power Economic Dispatch

Felix Calderon, Claudio R. Fuerte-Esquivel, Juan J. Flores, and Juan C. Silva

Universidad Michoacana de San Nicolás de Hidalgo
División de Estudios de Posgrado. Facultad de Ingeniería Eléctrica
Santiago Tapia 403 Centro. Morelia, Michoacán, México. CP 58000
calderon@umich.mx cfuerte@umich.mx,
juanf@umich.mx jcsilvach@yahoo.com

Abstract. This paper presents a new constraint-handling genetic approach for solving the economic dispatch problem in electric power systems. A real code genetic algorithm is implemented to minimize the active power generation cost while satisfying power balance (energy conservation) and generation limit constraints simultaneously during the optimization process. This is achieved by introducing a novel strategy for searching the solution on the energy conservative space, producing only individuals that fulfill the energy conservation constraint, and reducing the search space in one dimension. Computer simulations on three benchmark electrical systems show the prowess of the proposed approach whose results are very close to those reported by other authors using different methods.

1 Introduction

An electric power systems can be seen as the interconnection of generators and loads through a transmission network. The function of the transmission network is always to transport the electric energy from power plants to load centers. In this context, the Economic Dispatch (ED) problem is defined as the process of providing the required active power load demand and transmission losses by allocating generation among power plant units such that the total generation cost is minimized [1]. This goal is achieved if generator units are dispatching based on their function cost and generation limits while satisfying the total active power balance in the network. Hence, the ED can be stated as a constrained nonlinear optimization problem. Over the years, many classical optimization techniques have been used to solve this problem by generating a finite succession of test solutions, feasible or not, with search directions based on the gradient or superior order operators that produce solutions that converge to local optima (see [2]).

Evolutionary optimization techniques constitute an efficient and robust alternative to treat the ED problem. Among these techniques, Genetic Algorithms (GAs) basic principles were proposed in the 70's [3,4,5], and they have been applied for the ED problem for the last two decades. These algorithms are based

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 371–381, 2008.

on the natural evolution of species (Natural Selection)and offer many outstanding features over traditional optimization methods. For instance, their ability to search non-convex solution spaces with multiple and isolated maxima, global convergence, and inherent parallel search capability, among others characteristics [3].

In general, GAs applied to solve the ED problem use a set of random solutions associated to generator's active power. Each set constitutes a chromosome which is substituted in the objective cost function to compute the total active power generation cost. This process is repeated for each chromosome to assess the lowest production cost. Owing to the fact that GAs are unconstrained search techniques, methods based on penalty functions have been used to consider problem constraints during the optimization process. However, these methods require setting additional search parameters affecting the convergence performance. Another option is to preserve the feasibility of the solution by generating chromosomes that satisfy problem constraints. This can be achieved if it is possible to obtain the feasible solution region.

In this paper a constraint-handling GA approach is proposed based on preserving the feasibility of the ED solution. The proposed approach computes the feasible solution region by obtaining a vectorial search space based on the problem constraints. Results and analysis are presented to show the computational performance of the algorithm.

2 Economic Dispatch Problem Formulation

The ED problem is defined in terms of minimizing total active generation cost subject to satisfying the total active power demand and transmission losses, while maintaining the active power generation within limits [1]. For a system with N generation units, the cost $c(P_i, \alpha_i)$, for each generator is defined as a polynomial function given by Equation (1) where P_i is the active power for each generation unit, α_i is a set of polynomial coefficients, $\alpha_{i,j}$ is the $j-th$ polynomial coefficient associated to the $i-th$ generation unit and N_c is the polynomial order.

$$c(P_i, \alpha_i) = \alpha_{i,0} + \alpha_{i,1} P_i + \alpha_{i,2} P_i^2 + ... = \sum_{j=0}^{N_c} \alpha_{i,j} P_i^j \tag{1}$$

The total cost C is the sum of costs for each generation unit and the problem of minimizing the total cost can be formulated as an optimization problem (2) with restrictions (3) and (4).

$$Minimize \;\; C = \sum_{i=1}^{N} c(P_i, \alpha_i) \tag{2}$$

$$s.t \;\; \sum_{i=1}^{N} P_i = P_T \tag{3}$$

$$P_i^{\min} \leq P_i \leq P_i^{\max} \tag{4}$$

where P_T is the total active power demanded by loads and transmission system, and P_i^{min} and P_i^{max} the generation limits for each unit. Constraint given by Equation (3) corresponds to the total active power balance that must exists in the network, and it is referred to as Energy Conservation Constraint (ECC). On the other hand, constraints given by Equation (4) correspond to active power generation limits, referred to as Capability Constraints (CC).

For a quadratic cost, ($N_c = 2$), a unique minimum exists and the way to minimize it is by using Lagrange Multipliers [6]. Using Lagrange multipliers a new function, given by Equation (5), which includes the ECC constraint and lagrange multiplier λ, must be minimized. The most common method used in Power Engineering to solve (5) is the Newton approach.

$$\mathcal{L} = \sum_{i=1}^{N} \left[c(P_i, \alpha_i) + \lambda \left(\sum_{i=1}^{N} P_i - P_T \right) \right] \tag{5}$$

In the case of a non-quadratic system, the Newton algorithm does not guarantee to reach the global minimum and some authors as Wood and Wollenberg [7] apply an iterative strategy for computing the minimum, but this procedure does not guarantee to reach the global minimum; the solution depends on initial conditions. Other authors as Sheble, Orero, and Song [8,9,10] began to use searching strategies based on Simulated Annealing (SA) [11] and GA [4], which allows to find the global minimum for quadratic and non quadratic cost functions.

3 Economic Power Dispatch Using Genetic Algorithms

GAs are population based techniques which sample the solution space randomly by using several candidate solutions. These solutions are evaluated and those that perform better are selected to compose and generate the population in the next generation. After several generations, these solutions improve the objective function value as they explore the solution space for an optimal value.

In the context of ED problem, GAs optimize the cost function starting from a random population of possible solutions. Each individual solution is stored into a N-dimensional vector, whose $i-th$ floating point element or gene represents the active power output of the $i-th$ dispatchable generator. This data structure containing the active power outputs is known as chromosome. The fitness of a chromosome relates to the evaluation of the objective function and is a measure of goodness of this solution with respect to other solutions in the population.

Once the fitness of the initial population has been computed, a search for an optimal solution is carried out by updating the population according to rules of selection and stochastic operations called crossover and mutation. After performing a certain number of stochastic operations, some of the solutions in the old population are replaced by new solutions; this concludes one generation of the algorithm. Generations are repeated to minimize the total production cost until a stopping criteria is satisfied. In our case, the convergence criterion is given by the number of generation units.

Several authors have proposed GAs to tackle the ED problem. Sheble and Brittig [8] and Song and Chou [10] solve ED using GA. Both proposals consider a chromosome as a binary string representing all P_i's. The ECC (3) is handled as an error term and a measure of this error is calculated during the optimization process.

Orero and Irving [9] solve ED with prohibited operating zones using GA. They proposed to solve a non-continuous objective cost function given by Equation (6),

$$\hat{\mathcal{L}} = \sum_{i=1}^{N} \hat{c}(P_i) + \Psi \left(\sum_{i=1}^{N} P_i - P_T \right) + \Phi(P) \tag{6}$$

where $\hat{c}$ is a cost for the $i-th$ generation unit, Ψ and Φ are penalty functions for ECC and CC; respectively. This approach is not clear on how to select the Ψ and Φ functions; additionally chromosomes have binary representation.

Some Simulated Annealing (SA) based solution to the ED problem are presented by Wong and Fung [12], as well as Ongsakul and Ruangpayoongsak [13]. In [13], the purpose of this combination is to provide one initial solution for GA. It is used a quadratic cost function and 16 bits for chromosomes' binary representation.

A drawback of GAs using a chromosome's binary representation is that accuracy suffers of radical changes when the GA does not use the correct bit representation. For some cases, the chromosome length would have more bits than those needed by a floating point number representation in a computer (32 bits).

To avoid this problem, the proposed GA uses a real representation in order to increase the accuracy. Furthermore, instead of including the ECC into the cost function, a feasible solution region is generated by a linear combination of base vectors, such that feasible solutions are generated randomly within a so called vectorial search space to fulfill both ECC and CC.

4 Constraint-Handling Genetic Algorithm

The proposed approach to solve the ED optimization problem is described in this section. First, a reduction on the search space is accomplished through a set of vectors that define the intersection between a hypercube that represents CC, and the hyperplane that represents ECC. Next the GA solution based on those vectors is described.

4.1 Vectorial Search Space

Let $A = \{a_1, a_2, a_3, \ldots\}$ be a field of real number, whose elements will be called scalars. A vector space over the field A is a set V together two binary operation, vector addition and scalar multiplication. Two axiom are interesting for us: V is closed under vector addition (if $u, v \in V$ then $u+v \in V$) and scalar multiplication

(if $a \in \mathcal{R}$, $v \in V$, then $av \in V$); see [14]. Hence, any power vector $P(a) = [P_1(a), P_2(a), \ldots, P_N(a)]^T$ in the vector space ($P(a) \in V$) can be expressed as a linear combination of the vectors v_k making up the vector space, as given by (7). From (7) it is clear that the elements of a vector v_k corresponds to the on-line dispatchable active power generation.

$$P_i(a) = \sum_{k=1}^{M} a_k v_{k,i} \qquad i = 1, 2, \ldots, N \tag{7}$$

In the ED problem, the vector space needs to fulfill the ECC given by Equation (3). So it is necessary to compute the scalars a_k such that the $\sum_{i=1}^{N} P_i(a) = P_T$ and $\sum_{i=1}^{N} v_{k,i} = P_T \ \forall k$. Applying this condition to Equation (7) we obtain Equation (8).

$$\sum_{i=1}^{N}\sum_{k=1}^{M} a_k v_{k,i} = \sum_{k=1}^{M} a_k \sum_{i=1}^{N} v_{k,i} = \sum_{i=1}^{N} P_i(a) \tag{8}$$
$$\sum_{k=1}^{M} a_k P_T = P_T \Leftrightarrow \sum_{k=1}^{M} a_k = 1$$

From Equation (8) one can easily conclude that if the sum of the scalars is one, the linear combination of the vector will be a vector that fulfills ECC. Furthermore, if the active power generation representing by the elements of these vectors are within limits, the set of CC are also satisfied. Hence, the way of computing the vector collection v_k determines that both constraints could be fulfill simultaneously during the iterative process. The proposed approach to achieve this goal is described below.

In $\mathcal{R}^N$, the set of CCs are a hypercube whose corners are defined by the active power generation limits, and the set of ECCs constraint are a hyperplane. The main idea is to compute a vector set over the intersection between the hypercube and the hyperplane; the vector set can be created by a vector collection v_k, where each v_k corresponds to a vertex of the polygon that defines the intersection of CC and ECC. This condition reduces the search space in one dimension. Each v_k lies at the intersection between an edge of the hypercube and the hyperplane; for this reason v_k fulfills ECC – a desirable condition.

Given two vertexes, $P_m = [P_{m,1}, \ldots, P_{m,N}]^T$ and $P_l = [P_{l,1}, \ldots, P_{l,N}]^T$ in the hypercube CC, a parametric representation for edge $E_{m,l}(t)$ is given by Equation (9) with $t \in [0, 1]$.

$$E_{m,l}(t) = P_m + t\,(P_l - P_m) \tag{9}$$

There exists a value t^*, for the intersection between the hyperplane and the edge; such $\sum_{i=1}^{N} E_{m,l,i}(t^*) = P_T$, so if we add the components of edge $E_{m,l,i}(t)$, at position t^*, we can obtain an expression for t^* given by (10).

$$\sum_{i=1}^{N} E_{m,l,i}(t^*) = \sum_{i=1}^{N} P_{m,i} + t^* \sum_{i=1}^{N} (P_{l,i} - P_{m,i}) = P_T$$

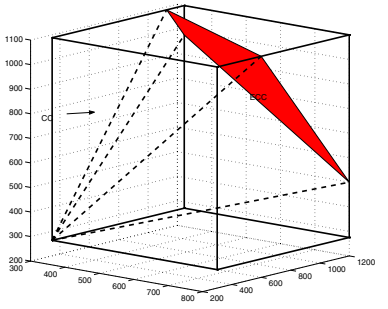


Fig. 1. Intersection between the CC and ECC in 3D

$$t^* = \frac{P_T - \sum_{i=1}^{N} P_{m,i}}{\sum_{i=1}^{N} (P_{l,i} - P_{m,i})} \tag{10}$$

If $t^* \in [0, 1]$ exists an intersection between edge $E_{m,l}(t)$ and the hyperplane ECC. So the corresponding vector $v_k = E_{m,l}(t^*)$ is accepted in the collection vector V. The procedure to compute the vector collection V is presented in Algorithm 1. The complexity for Algorithm 1 is lineal with respect the edge number, but the edge number is exponential with respect the dimension ($N \times 2^{N-1}$) and not necessarily exits a vector for each edge. Figure 1 shows the intersection of ECC (solid polygon) and CC (pipeline cube) in a three-dimension example. The doted lines correspond to vectors v_k from Wood network data presented in Table 2. Note that the search space instead of being a cube will be a plane in three dimensions, one less dimension. In this case the Generation units are $N = 3$, the number of vectors are $M = 4$ and the edges number are 8.

Algorithm 1. Vector searching Space

Given P_i^{min}, P_i^{max} and P_T
- set k=1 and $V = \{\emptyset\}$
- Compute de coordinates Q_m for each vertex of the Hypercube
- Compute the edges $E_{m,l}(t)$ of the hypercube
- For all edges $E_{m,l}(t)$
  * Compute t^* by Equation (10)
  * if $0 \leq t^* \leq 1$
    compute $v_k = E_{m,l}(t^*)$ by Equation (9) and $V \leftarrow V \cap \{v_k\}$
    $k \leftarrow k + 1$

4.2 GA Algorithm

Haupt [4] describes the steps to minimize a continuous cost function using GA. In our case, the chromosome instead of having a power representation, has a representation given by Equation (11), where each gene a_k is a floating-point

random number in $[0, 1]$ and will represent the scalar field to computing $P_i(a)$ by eq. (8). In order to fulfill the restriction imposed by Equation (8), the chromosome a is normalized using Equation (12). Given a chromosome a, the power vector $P(a)$ is computed by Equation (7) and the fitness function is given by Equation (13), which combines Equations (1) and (2).

$$a = [a_1, a_2, \cdots, a_k, \cdots, a_M] \tag{11}$$

$$a_k \leftarrow \frac{a_k}{\sum_{j=1}^{M} a_j} \tag{12}$$

$$C = \sum_{i=0}^{N} \left(\sum_{j=0}^{N_c} \alpha_{i,j} P_i^j(a) \right) \tag{13}$$

The initial population with size N_{pop}, after applying a sorting algorithm, is split in two halves. The best half is selected for mating (with $N_{good} = N_{pop}/2$ members); the other half will be discarded and replaced by the offsprings of the best half. Instead of using a random selection, a cost weighing function is used; the paring probability is computed using a normalized cost. The normalized cost for each chromosome is computed subtracting the best cost of the discarded chromosomes to the cost of all the chromosomes in the mating pool $\hat{C}_i = C_i - C_{N_{good}+1}$, $i \in [1, 2, \ldots, N_{pop}]$. The mating probability for each chromosome π_i is computed using Equation (14), so the best individual has the highest mating probability.

$$\pi_i = \frac{\hat{C}_i}{\sum_{j=1}^{N_{good}} \hat{C}_j} \qquad i = 1, 2, \ldots, N_{good} \tag{14}$$

Mating is carried out from a set of random couples. Given a couple of chromosomes p and m, a crossover point j is computed as a random integer number in $[1, M]$; the parameters around gene j are exchanged as shown by Equation (15). The genes at the $j-th$ position are combined using Equation (16), where β is a floating-point random number in $[0, 1]$.

$$O_1 = [p_1, \cdots, o_{1,j}, \cdots, m_M] \quad O_2 = [m_1, \cdots, o_{2,j}, \cdots, p_M] \tag{15}$$

$$o_{1,j} = m_j - \beta\,(m_j - p_j) \quad o_{2,j} = p_j + \beta\,(m_j - p_j) \tag{16}$$

Mutation is applied only to a percentage of the population, taking care of not mutating the best individual. So a random integer k is generated in $[2, N_{good}]$ and then a random integer j is generated in the interval $[1, M]$. The gene $a_{k,j}$, corresponding to the $k-th$ member, is replaced by a random floating-point number in $[0, 1]$. The new chromosomes generated by mating or mutation are normalized applying Equation (12). The procedure proposed in our approach (called GA-V) is shown in Algorithm 2.

5 Study Cases

The total power demanded for each network is 300.5576 KW, 2500 KW and 1443.4 KW, respectively; other data for each network are presented in

Algorithm 2. Genetic Algorithm GA-V

- Compute the vector collection V using algorithm 1
- Given N_{pop} and the mutation percentage, set $N_{good} = N_{pop}/4$
- Compute a initial population
  - for each population member
    - Create a chromosome $a = [0, 0, ...,]$ similar to (11)
    - Change randomly only M' genes with $M' \in [1, M]$
    - Normalize the chromosome using (12)
    - Evaluate the Power vector $P(a)$ by (7)
    - Given $P(a)$ and coefficients α_i evaluate the fitness function by eq. (13) and sort by fitness
    - Reduce the population size $N_{pop} \leftarrow N_{pop}/2$
- Repeat N_{gen}
  - * Compute the Mating probability using (14)
  - * Create random couples using the Mating probability and the best population half
  - * For each couple compute the offsprings by (15, 16)
  - * Replace the worst population half by the offsprings
  - * Mutate randomly the $k-th$ chromosome at the $j-th$ gene
  - * Normalize all the chromosomes by (12)
  - * Evaluate the fitness function (13) and sort by fitness
- For the best population member compute the power vector using (7)

Table 1. The first network contains five generators with a production cost given by a quadratic function cost ($N_c = 2$) whose coefficients are presented in column 2. The maximum and minimum limits for active power generation are given in columns 3 and 4, respectively. On the other hand, three generators are connected to the second and to the third network and the production cost is represented by a cubic cost function ($N_c = 3$). Data for generators of each network are also given in Table 1 in the same manner as indicated for the first network. The proposed GA approach (GA-V) was implemented using Java. The results of its application were obtained considering a population size of $N_{pop} = 1000$, number of generations $N_{gen} = 1000$, and mutation probability of 10%. These parameters were chosen in order to have a good performance for GA in the three cases.

The computed vectors that define the search space for these networks are presented in Table 2. It must be observed that each vector has $N-1$ active power generations at one of their limits, whilst the remaining is within limits. By way of example, for vector v_1 of the IEEE 14 system containing five generators, P_2, P_3, P_4 and P_5 generate at their maximum limit. On the other hand, generator P_1 generates 40.5576 MWs. The same observation applies to each vector v_k given in this table. Lastly, each vector satisfies $\sum_{k=1}^{N} v_k = P_T$, such that it is in $\mathcal{R}^N$; however in general the number of vectors v_k differs of the number or Generation units, as it is observed in table 2.

Table 1. System data

Network	Polynomial coefficients	Gen. limits	
		P_{max}	P_{min}
IEEE 14	$\alpha_1 = [0.02e-3, 0.003, 0.01]^T$	80.0	10.0
	$\alpha_2 = [0.02e-3, 0.003, 0.01]^T$	60.0	10.0
	$\alpha_3 = [0.02e-3, 0.003, 0.01]^T$	60.0	10.0
	$\alpha_4 = [0.02e-3, 0.003, 0.01]^T$	60.0	10.0
	$\alpha_5 = [0.02e-3, 0.003, 0.01]^T$	80.0	10.0
Wood	$\alpha_1 = [749.55, 6.95, 9.68e-4, 1.270e-7]^T$	800	320
	$\alpha_2 = [1285.0, 7.051, 7.375e-4, 6.453e-8]^T$	1200	300
	$\alpha_3 = [1531.0, 6.531, 1.04e-3, 9.98e-8]^T$	1100	275
Wong	$\alpha_1 = [11.200, 5.10238, -2.64290e-3, 3.33333e-6]^T$	500	100
	$\alpha_2 = [-632.000, 13.01000, -3.05714e-2, 3.33330e-5]^T$	500	100
	$\alpha_3 = [147.144, 4.28997, 3.08450e-4, -1.76770e-7]^T$	1000	200

Table 2. Searching space Vector computed by algorithm 1

Network	Active Power Vectors in KW
IEEE 14	$v_1 = [40.5576, 60.0000, 60.0000, 60.0000, 80.0000]^T$
	$v_2 = [80.0000, 20.5576, 60.0000, 60.0000, 80.0000]^T$
	$v_3 = [80.0000, 60.0000, 20.5576, 60.0000, 80.0000]^T$
	$v_4 = [80.0000, 60.0000, 60.0000, 20.5576, 80.0000]^T$
	$v_5 = [80.0000, 60.0000, 60.0000, 60.0000, 40.5576]^T$
Wood	$v_1 = [320.0, 1080.0, 1100.0]^T$
	$v_2 = [800.0, 600.0, 1100.0]^T$
	$v_3 = [320.0, 1200.0, 980.0]^T$
	$v_4 = [800.0, 1200.0, 500.0]^T$
Wong	$v_1 = [343.40, 100.00, 1000.00]^T$
	$v_2 = [100.00, 343.40, 1000.00]^T$
	$v_3 = [100.00, 500.00, 843.40]^T$
	$v_4 = [500.00, 500.00, 443.40]^T$
	$v_5 = [500.00, 100.00, 843.40]^T$

Table 3. Comparative Results

Network	Algorithm	Demand	Generation Cost	Time (seg.)	Active Power Vector Solution
IEEE 14	[13]	300.5576	181.5724	-	[60.2788 60.0000 ... 60.0000 60.0000 60.2788]
	GA-V	300.5576	181.5724	4.276	[60.2789 59.9999 ... 59.9999 59.9997 60.2789]
Wood	[7]	2500.1000	22730.21669	-	[726.9000 912.8000 860.4000]
	GA-V	2500.0000	22729.32458	2.814	[725.0078 910.1251 864.8670]
Wong	[12]	1462.4480	6639.50400	-	[376.1226 100.0521 986.2728]
	GA-V	1443.4000	6552.23790	2.842	[343.3980 100.0415 999.9604]

5.1 Numerical Results

A comparison of the results obtained by the proposed approach GA-V, with respect to those reported by other authors are shown in Table 3. This table shows the algorithm used, the power demand, the computed generation Cost, the time required to compute the solution using GA-V (Algorithm 1 and 2) and the final generator active power vector. We select the IEEE14 and Wong networks because their solution are obtained by an algorithm based on Simulated Annealing in [13] and [12]. Results obtained for the IEEE14 system are identical to those reported by Ongsakul [13]. For the second and third networks, GA-V yields better solutions than those reported in [12] and [7], respectively. Additionally to the improvement of the generation cost, it must be pointed out that the solutions reported by Wood and Wong do not fulfill the energy conservation equations, exactly. This is a necessary condition to a feasible steady state operation of any electric power system.

A comparison in computational time is not possible because it was not reported in [7], [12] and [13]. However, computational times for the networks solution based on the proposed approach is also reported in Table 3. This information provides an idea of the algorithm complexity.

6 Conclusion

In this work, a new strategy called vector search space has been proposed for stochastic search of the active power dispatch that produce the lowest generation cost. This strategy guarantees that generation dispatch fulfill the restrictions imposed by the ED problem, reducing the search space in one dimension, therefore reducing the convergence time.

One drawback of GA-V is that it needs to compute the vectors that intersect the edges of the hypercube with the hyperplane. In $\mathcal{R}^N$ the number of edges is $N \times 2^{N-1}$, so its complexity is exponential. In future works we will try to devise a strategy to compute the vector collection in polynomial time.

References

1. Stevenson, W.D.: Elements of Power System Analysis. McGraw-Hill, New York (1982)
2. Momoh, J.A.: Electric Power System Applications of Optimization. CRC Press, Boca Raton (2005)
3. Pohlheim, H.: Evolutionary Algorithms. Springer, Heidelberg (2003)
4. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms, 2nd edn. John Wiley and Son, Inc., New York (2004)
5. Goldberg, D.E.: Genetic Algorithms. Addison-Wesley, New York (1989)
6. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
7. Wood, A., Wollenberg, B.: Power Generation, Operation, and Control. John Wiley and Sons Inc., Chichester (1984)

8. Sheble, G.B., Brittig, K.: Refined genetic algorithm-economic dispatch example. IEEE Transactions on Power Systems 10, 117–124 (1995)
9. Orero, S.O., Irving, M.R.: Economic dispatch of generators with prohibited operating zones: A genetic algorithm approach. In: IEE Proceedings. Generation, transmission and distribution, Institution of Electrical Engineers, vol. 143, pp. 529–534 (1996)
10. Song, Y., Chou, C.: Advanced engineered-conditioning genetic approach to power economic dispatch. In: Generation, Transmission and Distribution, IEE Proceedings, vol. 144, pp. 285–292 (1997)
11. Aarts, E., Korst, J.: Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing. John Wiley, Chichester (1991)
12. Wong, K., Fung, C.: Simulated annealing based economic dispatch algorithm. In: Generation, Transmission and Distribution, IEE Proceedings C, vol. 140, pp. 509–515 (1993)
13. Ongsakul, W., Ruangpayoongsak, N.: Constrained dynamic economic dispatch by simulatedannealing/genetic algorithms. In: 22nd IEEE Power Engineering Society International Conference on Power Industry Computer Applications, 2001. PICA 2001. Innovative Computing for Power - Electric Energy Meets the Market, pp. 207–212 (2001)
14. Strang, G.: Linear Algebra and Its Applications, 3rd edn. Thomson Learning, USA (1988)

Using Genetic Fuzzy Algorithms to Model the Evolution of Climate Variables at San Jorge Gulf Area

Enrique Sierra, Daniel Lorenzetti, Alejandro Hossian, Hugo Moyano, Horacio León, Daniel Pandolfi, Rodolfo Gómez, and Ramón García Martínez[†]

Laboratorio de Energías Renovables. Unidad Académica Caleta Olivia
Universidad Nacional de la Patagonia Austral
Ruta Nacional Nº 3 Acceso Norte (9011) Caleta Olivia- Santa Cruz – Argentina
Telefax: 54-297-4854-888
enriquesie@yahoo.com.ar, dloren@uaco.unpa.edu.ar
http://www.unpa.edu.ar

[†] Centro de Ingeniería del Software y del Conocimiento Instituto Tecnológico de Buenos Aires.
Laboratorio de Sistemas Inteligentes. Facultad de Ingeniería.
Universidad de Buenos Aires.
rgm@itba.edu.ar, rgarciamar@fi.uba.ar
http://www.centros.itba.edu.ar/capis/ -
http://laboratorios.fi.uba.ar/lsi/index.htm

Abstract. The following paper presents a genetic fuzzy model that utilizes special features, for the prediction of the wind speed variations in various time windows in the future. The model utilizes a set of meteorological stations that encircle the wind turbine clusters at a radius of 15km or more. The system has been applied using data collected over a period of 2 years at locations in the northern part of South Patagonia, Argentina, in the area called San Jorge Gulf. It achieves an understanding of the problem autonomously. A user intervention to assist the training process is not needed while there is also no need for a certain parameters' initialization, like in other methods, which contributes to the model's robustness.

Keywords: Genetic Algorithms, Fuzzy Logic, Climate, Modeling.

1 Introduction

Large rapid changes in wind power can cause serious reduction in operating economy and reliability, especially for utilities with large wind generation penetration like in northern region of South Patagonia, Argentina. Prediction of wind power, along with load forecasting, can allow a utility to schedule the connection or disconnection of wind or conventional generation, achieving low spinning reserve at optimal operating cost. In the past, efforts have been made to simulate wind power variations, mostly by analyzing the wind-speed time-series of a certain site. Unfortunately, the statistical properties of wind-speed are not very helpful. Therefore, prediction for the next 15 minutes to one hour was practically close to persistent forecast, i.e. suggests no change from the most recent values.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 382–393, 2008.

On this basis, authors often turned to spatial correlation studies of wind speeds, not always leading to a satisfactory model. A significant correlation of hourly or daily average speeds is certain for distances of 20 to 100 km. It is suggested that correlation decreases with distance [3] and topographical elevation difference [1]. It also decreases when the orientation of the distance vector differs from the wind direction [11] [16]. Studies have been made even for spatial correlation of wind turbulence regarding short distances (700 m to 15 km) and short time scales (wind changes per 4, 10, 30 min and also 1-minute deviations from 30-min averaged value). Correlation coefficients are concluded to be related to the wind direction, terrain roughness and height above the ground [2] [17] [18]. More significant are Schlueter's two projects since they present complete correlated echelon models [12] [13]. Local wind data are not exploited though and the model accuracy is not evaluated or checked out on future events.

The present paper induces a genetic fuzzy system that uses local and spatial relations of the wind speed so as to improve the efficiency of forecasting, ranging from minutes to several hours ahead. Since the efficiency of a genetic algorithm (GA) based solution depends greatly on the coding scheme, the proposed GA for the training of the fuzzy system uses an effective coding scheme as well as special operators, tailored to the problem. The model is flexible and self-sufficient so as to easily learn and adapt in data of any special case without any guidance or constraint by the user. Tests are supported by measurements at several terrains during a period of two years.

1.1 Problem Formulation

Let us consider two sites A (upwind) and B (downwind) laying in the direction of the wind. At a sufficient height above the ground we can accept there is a uniform wind speed field. This is valid for the boundary layer, where winds are caused by global pressure gradients and remain unaffected by heat transfer and terrain features. For a considerably short distance (AB) we may denote the uniform wind speed, above the two sites as follows:

$$v_A(t) = v_B(t) \tag{1}$$

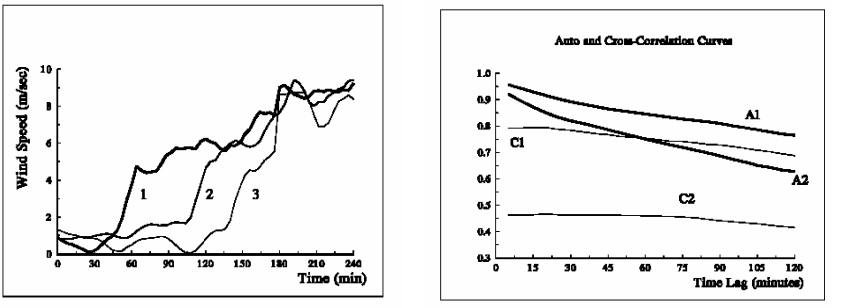
Any meteorological event arriving at A on time t = 0, may be realized as a distinct change Δv in the wind speed. This event is assumed to propagate with speed $v_A(t)$ and is expected to arrive at B after a delay time proportional to $\Delta\tau$:

$$\Delta\tau = \frac{AB}{v_A(t)} \tag{2}$$

This delay time is obviously different for each case, it depends on the wind speed and direction. Close to the surface (15 or 30 m above the ground), wind field can not be accepted as uniform. The propagation of a disturbance is possible to be followed by solving the equations governing the flow (e.g. Navier - Stokes) taking into account the pressure gradient, the boundaries of the terrain and possibly heat transfer. This method is rather complex and its efficiency is doubtful. However, we may assume that there is some law of propagation in every case, and a weak or strong relation exists between speeds at A and B, sited at a certain terrain. The object of research is to

arrive to a model that describes and forecasts the future small or large fluctuations of wind speed at site B, based on recent variations of wind speed as registered at sites in the neighborhood A_j with j=1…NS.

Watching the wind profiles at sites Aj and B a relationship, more or less intense, is already established. It seems that speeds v_{Aj} are projected at B with a varying time delay and a certain change at their magnitude Figure (1a). Time series analysis of wind speed can give the average $\mu_{Aj,}$ $\mu_{B,}$ the standard deviations σ_A, σ_B of the speeds and the cross correlation coefficients $\rho_{ABj}(\tau)$. The case used now as an example is the first one of the application cases that will be described extensively later on. The prevailing state is that v_B follows v_{Aj}.



(a) (b)

Fig. 1. (a) Typical view of an oncoming front as measured at three sites 1, 2 and 3. The sites are lying along a line 40-km long on the direction of the propagation of the front. Distances are (1-2) = 27 km, (2-3)=12 km. (b) Auto (A) and cross-correlation (C) curve for wind speeds at two sites 12 km apart. Two pairs of curves are presented concerning: (1) strong NW winds and (2) low winds of all directions.

For instance, we may assume that:

- When winds at A_j are strong and parallel to (A_jB) axis there is a high correlation of v_{Aj} and v_B with a time delay of about 20 to 30 min.
- When winds at A_j are low then the winds at B are also low but there is no particular connection in their fluctuations. This is because weak winds are affected by ground morphology and the phenomenon recorded at the remote site deteriorates by the time it reaches the local site.
- When winds are at least medium and they are vertical to (A_jB) axis there is an unconsidered time delay since the front arrives at the sites almost at the same time.

This is clear from cross-correlation curves extracted separately by patterns of data that belong to each one of the above cases. Therefore, it is acceptable to have a clustering of these cases and then specify the v_{Aj} / v_B relation for each case. Such a structure reminds us of the quite familiar inference if-then rules [15] that have been successfully used in several fuzzy systems applications the recent years.

2 The Fuzzy Wind Forecasting Model

Fuzzy logic is used where a system is difficult to model precisely (but an inexact model is available), or where ambiguity or vagueness is common [7][19]. Based on space's fuzzy clustering the model:

a) sets k different cases or rules. Each one of them has a specific input - output function.
b) takes every input-output pattern and determines its degree of participation in each one of the rules
c) optimizes the input - output function (which in our model is a linear one) for each one of these rules (training process).

Each fuzzy rule consists of two parts, the *premise* part and the *consequence* part. N premise inputs $x_{p1},...,\ x_{pN}$ have to be selected. These are the decision variables that create the N-dimensional input space, which is going to be partitioned in fuzzy subspaces.

In our problem, two premise inputs are chosen:

• **The wind speed** of what is supposed to be the oncoming event. This is the wind speed as measured at the most remote 'upwind station'. For a reference site B literally surrounded by remote stations A_j, we call 'upwind station' the one for which the axis (A_jB) is closest to the direction of the wind, i.e.:
- The Eastern station if an Eastern wind is measured there
- The Southern station if a Southern wind is measured there, etc.

• **The direction** of the above mentioned wind.

These premise values of wind and direction must be averaged within a time range of 15 to 30 min, so that
a) Turbulence is extracted
b) The information about the most recent changes remains

In practice, the dominant wind directions we are interested in, may be only two or even one. In each case the remote stations must be sited properly, as close as possible to the axis of the direction of the fronts. For the division of each dimension, 3 gaussian membership functions are used forming 9 fuzzy if-then rules. For instance, for wind speed premise input x_{p1}, the 3 membership functions $MF_1(x_{p1})$, $MF_2(x_{p1})$, and $MF_3(x_{p1})$ may be used to express linguistic properties like 'wind is weak', 'wind is medium', and 'wind is strong'. So, if a specific sample $x_{p1} = 5$ m/s gives:

$$MF_1(x_{p1}) = 0.4 \quad MF_2(x_{p2}) = 0.7 \quad MF_3(x_{p1}) = 0.05$$

This would state that x_{p1} is rather attributed as a weak-medium wind. In the consequence part, the output function of each fuzzy rule j is defined. This is a linear function of the consequence inputs and it is shown below [14]:

$$y_j = f_j\left(x_{c1}, x_{c2}, ...x_{cM}\right) = \lambda_0 + \sum_{i=1}^{M} \lambda_i x_{ci} \tag{3}$$

Inputs $x_{c1},...,x_{cM}$ are the so called consequence inputs that form the input-output patterns. The consequence inputs may be many (even close to a hundred). These are recent wind speeds from the local site B and the remote sites Aj, the speeds presented below:

$$y_I = \frac{\sum_{j=1}^{NR} \mu_{I_J} y_j}{\sum_{j=1}^{NR} \mu_{I_J}} \tag{4}$$

So, for each pattern (I=1 to the number of patterns NP) we have its degree of participation μ_{IJ} in each fuzzy rule j (j=1 to the number of rules NR). Then the final output y_I is given as the weighted average of all rules outputs y_j.

A genetic algorithm solution is used in order to adjust the premise and consequence parameters as it will be described in the next section. The premise parameters are the center, and the spread of each gaussian membership function, while the consequence parameter are the λ_{ci} of the rules. The aim of the training is to minimize the error during the training of the fuzzy system using a training input-output database.

3 Genetic Algorithm for the Training of the Fuzzy Wind Forecasting Model

Genetic Algorithms (GAs) are conceptually based on natural genetic and evolution mechanisms working on populations of solutions in contrast to other search techniques that work on a single solution [8]. The most interesting aspect of GAs is that although they do not require any prior knowledge or space limitations such as smoothness or convexity of the function to be optimized, they exhibit very good performance on the majority of the problems applied [4].

At first a population of i.e. M chromosomes (genotypes) is generated at random. In order to train the fuzzy system using a genetic algorithm as a training method, we have to encode the fuzzy system's parameters into a chromosome. This chromosome will be the genetic material that the algorithm will optimize through evolution. For the formation of the chromosome and the encoding of the parameters, a binary coding was selected. This was done primarily for algorithmic simplicity and furthermore for the implementation of some special operators that will be described later in this paragraph.

Eight bits per parameter were used. This coding scheme offers a resolution of 256 discrete values which is considered satisfactory; 16, 32 or more bits could be used in order to encode each fuzzy system's parameter and as a result the resolution would be much greater and the coded parameter closer to its real value. On the other hand though, the chromosome length would increase in a great extent, and the training would be much slower, while it is doubtful whether the results would be better, since

the main advantage of a fuzzy system is that it doesn't need exhausting accuracy in order to recognize a pattern.

In that way the parameters are encoded and placed one after the other to form a chromosome as it is shown in the following figure, Figure 2.

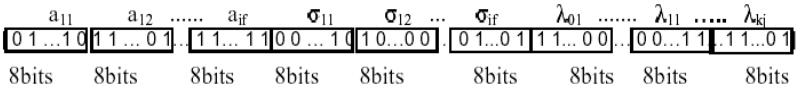


Fig. 2. Chromosome structure with the allocation of rules' parameters

The parameters of the fuzzy system that must be coded are as we saw in previous section of three kinds:

1. The a_{if} factors, which are the centers of the membership functions that cover the premise space.
2. The σ_{if} factors, which are the standard deviations of the membership functions, providing that we use Gaussian type membership functions.
3. The λ_{kj} factors, which define the output of each fuzzy rule, as a function of the consequence part inputs.

After the creation of the initial population, each of the chromosomes is evaluated. This is done in three steps:

a) the chromosome is decoded by breaking into 8 bit pieces and each of these segments represents a parameter of the fuzzy system as it was described above.

b) having decoded the parameters the fuzzy system's output for each of the patterns is calculated.

c) the average and mean square error for all the patterns of the training database are calculated. The reverse mean square error is used as the fitness function for the evaluation of the chromosome:

$$Fitness\ Function = \frac{1}{\sum_{1}^{NPATTERNS} \frac{1}{2}\left[y^{I}_{REAL} - y^{I}_{FUZZY}\right]^2}$$

where:

y^{I}_{REAL} is the wind speed in the specified window in the future as was taken by the measurement data.

y^{I}_{FUZZY} is the fuzzy system's output (prediction) for pattern I based on the system's current training. When $y^{I}_{REAL} - y^{I}_{FUZZY}$ the fitness function equals one.

As we can see, a chromosome's fitness maximizes when the mean square error becomes minimum, meaning that the fuzzy system represented by the chromosome has acquired a total understanding of the problem and its predictions are very close to the real values.

After the evaluation is completed, genotypes from the population are selected in pairs to replicate and form «offspring» genotypes. The selection of the genotypes is performed probabilistically using the **Roulette Wheel Parent Selection Algorithm [4]**, which selects genotypes with a probability proportional to their relative fitness within the population. This assures that high-quality solutions will be selected many times to become the «parents» of many new solutions.

When two parent genotypes are selected their binary strings are combined to produce an «offspring» solution using genetic-like operators. The main operators used are **Crossover** and **Mutation**.

- **Crossover operator** is applied with a probability that adjusts according to the convergence of the population. In that way, when excessive convergence is noted the crossover probability is lowered by 10%, while in cases of excessive divergence the probability increases. A multi-point crossover scheme (shown in Figure 3a) with 6 crossover points was used for our problem.
- **Mutation operator** (Figure 3b) is applied with a small probability. This probability adjusts to the convergence of the population in a similar way to the crossover probability. Because mutation favors population's diversity, the mutation probability is raised when excessive convergence occurs, while it is lowered when the population shows excessive diversity. Randomly chosen bits of the offspring genotypes flip from 0 to 1 and vice versa to give characteristics that do not exist in the parent population.

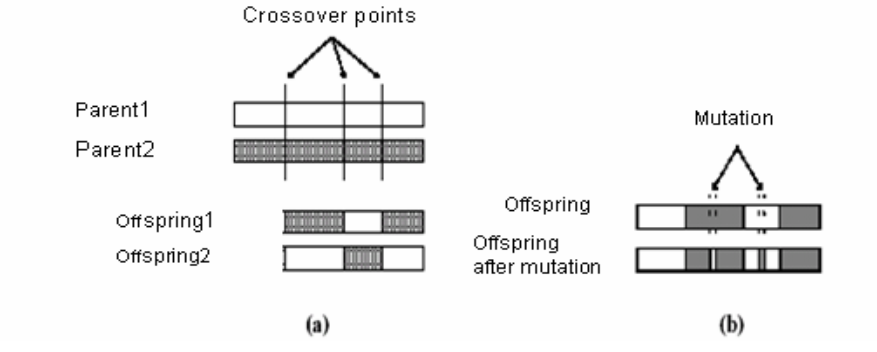


Fig. 3. a) Multipoint crossover scheme b) Mutation operator

In addition, other genetic operators and techniques are used to improve the performance of the GA, such as **Elitism** and **Fitness Scaling**.

- **Elitism** is implemented so that the best solution of every generation is copied to the next so that the possibility of its destruction through a genetic operator is eliminated.
- **Fitness Scaling** is referred to a nonlinear transformation of genotype fitness in order to emphasize small differences between near optimal qualities in a converged population. In addition a new operator tailored to the specific problem was developed, which in conjunction with the coding is very effective in helping the system avoid premature convergence and in addition reduces the training time significantly. This operator is described below in detail.

• **Consecutive Variable Swapping Operator (CVSO)**

This special operator is applied to the bits of every chromosome with a fixed probability 10% for each chromosome. The operator is implemented into 2 phases and is shown in Figure 4. Initially two variables (bit substrings representing fuzzy variables) are chosen randomly within the chromosome. Then the bits of the first variable are copied to the bits of the second variable and the fitness value of the new chromosome is evaluated. If it is greater than the fitness value of the initial chromosome then the new chromosome replaces the old one inside the population. If not the whole process is reversed. The bits of the second variable are copied to that of the first, the new fitness value is evaluated and if it is greater than the old one a replacement of the chromosome is done. If none of the above phases give a better fitness the chromosome is left unchanged.

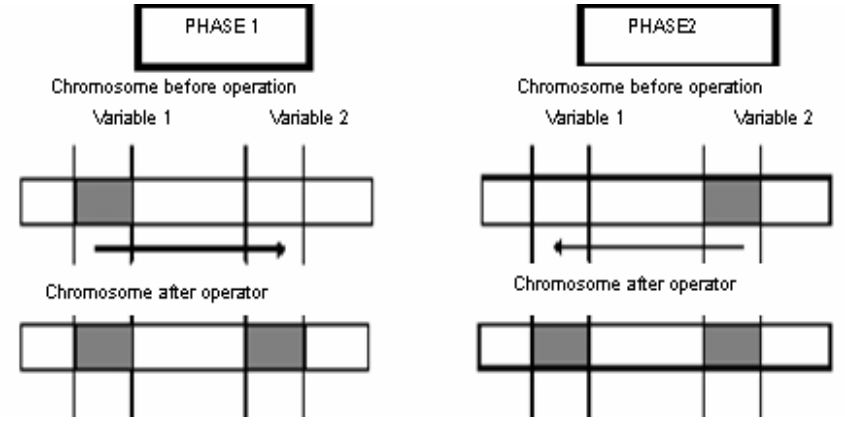


Fig. 4. Consecutive variable swapping operator

The power of the new operator comes in combination to the coding scheme used for the fuzzy parameters to which the operator is applied. Considering the nature of the problem we can conclude that the future wind speed at the local station is the result and the sum of all the past wind speeds at the local and distant stations. The parameters set the degree to which each of these speeds contributes to the speed we want to calculate. From the above we can conclude that these parameters must be positive and not negative values.

The use of the CVSO operator has as result the copying of one bit string to another. When two variables are chosen randomly, the one of course is smaller than the other so when it is copied to the other the fitness value is almost certainly better.

With the use of this operator the smaller variables reproduce in the chromosome with almost geometrical rate while the fitness value grows equally fast. Since almost all factors are at the optimal solution very small, this operator decreases the training time in a great extent and helps locating an optimum solution relatively fast. When M new solution strings are produced, they are considered as a new generation and they replace the parents in order for the evolution to proceed. Many generations are needed for the population to converge to the optimum or a near-optimum solution and the number is increasing according to the problem's difficulty.

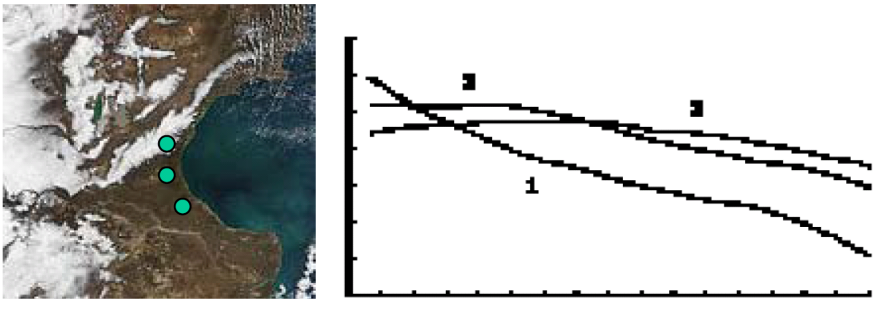
4 Test Results

In order to test the efficiency of the GA to Fuzzy Wind Forecasting problem, many benchmark tests have been carried out using real data measured from the San Jorge Gulf area using 3 stations. The locations of the local and 2 remote stations are shown in Figure 5a. The stations were installed in 10m height and away from any nearby buildings. Since between the stations there is smooth ground or sea we can assume that the wind phenomena travel with steady speed and that there is minimum alteration. The distances between the stations are: (A_1A_2) = 27 km, (A_1B) = 58.9 km, (A_2B) = 31.9 km. Station B was used as the local station for testing of model quality. The gathering of the data was made using 3 personal computers and lasted a year. The main wind phenomena in the great area of San Jorge Gulf are the following:

a) Strong winter winds with north-northwest direction that can reach speeds of up to 25 m/s. These winds appear mainly during the night and their direction is parallel to the (AB) axis of the stations.
b) An aura which is noticeable mainly during the summer time. These are winds that reach speeds of up to 8.8 m/s and they appear only during the day.

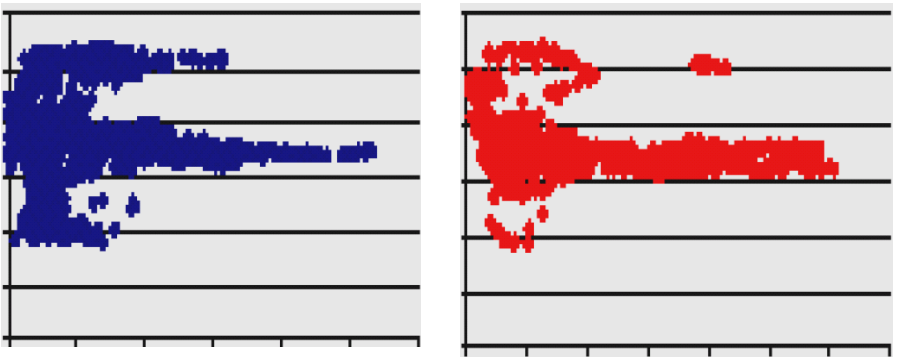
The correlation factor between the wind speeds of the local and the remote stations is big as Figure 5b shows. For long forecasting horizons data from the distant stations are more related. We can see that the correlation curves have nodes for the points Δt = 15, 25, and 55 min. From this observation we could conclude that for forecasts of 25 minutes ahead and longer the data from the distant stations could prove more important than the local station data.

Figure 6a shows the patterns' positions in the premise space defined by 2 premise inputs: the wind speed and the wind direction. We can see that speeds cover the range from 0 to 13m/s while direction shows peaks at the northwest (~350 degrees) and south (~500 degrees) directions. Figure 6b shows the patterns for predictions.
Figure 7 shows the positioning of the membership functions in the premise space after training. The fuzzy system has achieved a good understanding of the model, since the functions are located in such places that cover all the range of speeds, and the peaks at northwestern, and southern directions that appear in the wind direction input space.



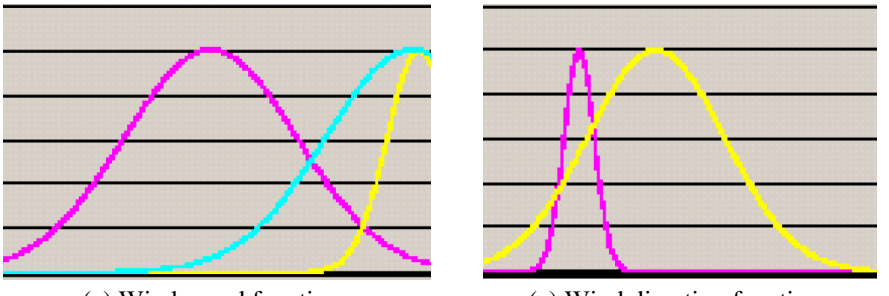
(a) Location map b) Correlation curves

Fig. 5. Location map (a) and Correlation curves (b)



(a) Training patterns (a) (b) Prediction patterns

Fig. 6. Training and prediction patterns locations in the models premise space



(a) Wind speed functions (a) Wind direction functions

Fig. 7. Membership functions for wind speed (a) and wind direction (b) for the two premise inputs fuzzy model

5 Conclusions

A genetic algorithm trained fuzzy model was developed in order to predict wind speed and hence power production of wind generators. The model was tested in many sites using data collected over a period of two years. Even though the test sites present very different meteorological characteristics the genetic fuzzy model achieved a very good performance in all cases. This was owed in a great extent to the new operator that was developed for the needs of the problem and had as a result the reduction of training time, as well as the improvement of the optimum solution quality. Because the fuzzy system is trained for the data of two years for each site it can be trained once and then be used for online prediction which is done with the delay of less than a second. In this way the main disadvantage of the genetic algorithms is overcome. This research work has been fully financed by a grant issued by Unidad Académica Caleta Olivia, Universidad Nacional de la Patagonia Austral, Santa Cruz, Argentina.

References

1. Beyer, H.G., Luther, J., Steinbergerwillms, S.: Power Fluctuations in Spatially Dispersed Wind Turbine Systems. Solar Energy 64(4), 297–305 (1993)
2. Chan, S.M., Powell, D.C., Yoshimura, M., Curtice, D.H.: Operations requirements of utilities with wind power generation. IEEE Transactions on Power Apparatus & Systems PAS-102(9) (1983)
3. Corotis, R., Sigl, A., Cohen, M.: Variance Analysis of Wind Characteristics for Energy Conversion. Journal of Applied Meteorology 16, 1149–1157 (1977)
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimization & Machine Learning. Addison Wesley, New York (1989)
5. Herrera, F., Verdegay, J.L. (eds.): Genetic Algorithms & Soft Computing. Springer, New York (1996)
6. Sierra, E., Hossian, A., Britos, P., Rodríguez, D., García Martínez, R.: Fuzzy Control for improving energy management within indoor building environments. In: Proceedings CERMA Electronics, Robotics & Automative Mecahnics Conference. IEEE Computer Society Press, Cuernavaca (2007)
7. Holland, J.H.: Outline for a logical theory of adaptive systems. Journal Ass. for Comput. Mach. (ACM) (1962)
8. Holland, J.H.: Adaptation in Natural & Artificial Systems. Ann Arbor MI/USA Mich. Univ. Press (1975)
9. Jang, S.R.: ANFIS: Adaptive-Network-Based Fuzzy Inference Systems. IEEE Transactions on systems, Man and Cybernetics 23(3), 665–685 (1993)
10. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn. Springer, Heidelberg (1996)
11. Palomino, I., Martin, F.: A Simple Method for Spatial Interpolation of the Wind in Complex Terrain. Journal of Applied Meteorology 34(7), 1678–1693 (1995)
12. Schlueter, R.A., Park, G.L., Bouwmeester, R., Shu, L., Lotfalian, M., Rastgoufard, P., Shayanfar, A.: Simulation and Assessment of Wind Array Power Variations Based on Simultaneous Wind Speed Measurements. IEEE Transactions on Power Apparatus and Systems PAS-103(5), 1008–1016 (1984)
13. Schlueter, R.A., Sigari, G., Costi, A.: Wind Array Power Prediction For Improved Operating Economics And Reliability. IEEE Transactions on Power Apparatus and Systems PAS-104(7), 137–142 (1985)
14. Takatsi, T., Sugeno, M.: Fuzzy identification of systems and its Applications to Modeling and Control. IEEE Transactions on systems, Man and Cybernetics SMC(1), 116–132 (1985)
15. Sierra, E., Hossian, A., Rodríguez, D., García Martínez, M., Britos, P., García Martínez, R.: Optimizing Buildings' Environment Performance Using Intelligent Systems. In: Nguyen, N.T., Borzemski, L., Grzech, A., Ali, M. (eds.) IEA/AIE 2008. LNCS (LNAI), vol. 5027. Springer, Heidelberg (2008)
16. Kodama, N., Matsuzaka, T.: Web-Based Data Acquisition System of Wind Conditions and its Application to Power Oput Variation Analysis for Wind Turbine Generation. In: SICE-ICASE International Joint Conference, pp. 3747–3750 (2006)
17. Anderson, C., Cardell, J.: Reducing the Variability of Wind Power Generation for Participation in Day Ahead? Electricity Markets. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, p. 178 (2008)

18. Jursa, R.: Variable selection for wind power prediction using particle swarm optimization. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 2059–2065 (2007)
19. Jaya Suma, G., Shashi, M., Lavanya Devi, G.: A Comprehensible Approach to Develop Fuzzy Decision Trees Foundation. In: First International Conference on Emerging Trends in Engineering and Technology, pp. 601–605 (2008)

Digital Filters Using Adjacency Matrix Representation

Leonardo Sá and Antonio Mesquita

Abstract. An evolutionary algorithm able to synthesize low-sensitivity digital filters using a chromosome coding scheme based on the adjacency matrix is proposed. It is shown that the proposed representation is more flexible than GP-tree schemes and has a higher search space dimension.

Keywords: Circuit synthesis, digital filters, evolutionary computation, genetic algorithms.

1 Introduction

Digital filter synthesis is the process of converting a transfer function into a filter structure composed of three basic elements: delays, multipliers, and adders. There are several filter structures able to realize the same transfer function. The complexity and the number of elements of these structures can vary significantly depending on the synthesis method. In this paper, an efficient evolutionary synthesis method for low-sensitivity digital filters is proposed. It is shown that the effectiveness of the proposed method derives mainly from the adjacency matrix chromosome representation used to describe the filter structure during the evolutionary process. It is also shown that the proposed chromosome coding scheme is more flexible than the GP-tree representation [1-2]. In fact, the proposed representation can map a wider number of topologies than can be achieved by GP-based evolutionary algorithms.

Among the chromosome coding schemes proposed to represent circuits in evolutionary synthesis methods, those based on GP-tree have deserved the researcher's preference [3-4]. This is mostly due to the possibility of representing multi-terminal components in a straightforward way when dealing with low level synthesis. Nevertheless, graphs are the most convenient data structure to represent circuit's topologies. However, due to the difficulty of representing multi-terminal components, evolutionary methods using chromosome coding schemes based on graph matrices are suitable only for high level synthesis using two-terminal functional blocks [5]. However, it will be shown that, in the particular case of the digital filters synthesis, this limitation can be removed from the graph-based chromosome coding schemes in such a way that they can be efficiently used.

In a digital filter, the adders, having two input terminals and one output terminal, are the sole multi-terminal element and, as will be shown, they can be unambiguously described by an oriented graph composed of two independent branches.

As a further advantage of the proposed chromosome coding scheme, the detection of filters containing delay-free loops that affect considerably the efficiency of any evolutionary synthesis method, is easily implemented with only a small amount of computation.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 394–406, 2008.

The content of the paper is outlined as follows. The proposed adjacency matrix representation for digital filters is discussed in Section 2. The rules to construct valid individuals including the computability are addressed in Section 3. The application of the genetic operators and a comparison with the GP-tree representation are discussed in Sections 4 and 5, respectively. The heuristic of the fitness computation for a second-order low-sensitivity digital filter synthesis is described in Section 6. Concluding remarks are given in Section 7.

2 Adjacency Matrix Representation for Digital Filters

Let $G(V,E)$ be an oriented graph with no parallel edges and n vertices sorted out between v_1 and v_n, where V and E denote, respectively, the set of vertices and edges of the graph. The adjacency matrix $A=[a_{ij}]$ of the oriented graph G is the $n\ x\ n$ matrix defined as [6]:

$$a_{ij}=\begin{cases}1, & \text{if } (\mathrm{v_i},\mathrm{v_j}) \in \mathrm{E}\\ 0, & \text{otherwise}\end{cases} \tag{1}$$

An edge is said to be incident out of its initial vertex and incident into its terminal vertex. A vertex is called an isolated vertex if no edge is incident on it. In Fig. 1(a) the edges corresponding to the graph G of Fig. 1(c) are indicated in the adjacency matrix A. In Fig. 1(b) the edges are replaced by 1s according to (1). In this matrix the non-zero main diagonal entries represent self-loops or short-circuits as, for example, the a_{33} entry in Fig. 1(b) that corresponds to the edge e_7 in Fig. 1(c).

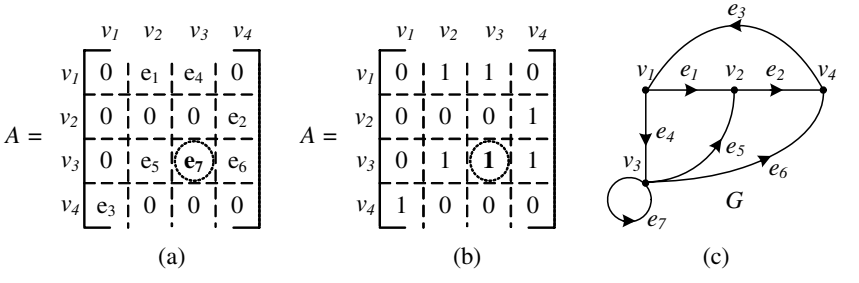
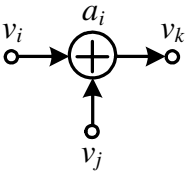
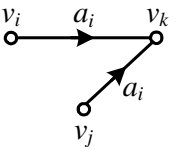
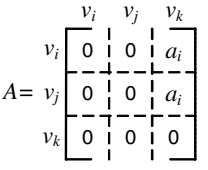
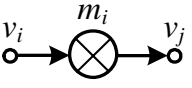
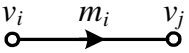
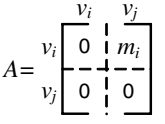
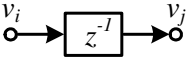
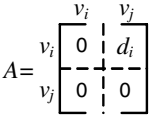


Fig. 1. (a) Adjacency matrix using edges (b) adjacency matrix (c) corresponding oriented graph G

The adjacency matrices corresponding to the three basic elements of a digital filter are given in Table 1. As shown in the table, the problem associated with the representation of adders is solved by a two-edge oriented graph. As a consequence, due to the three-terminal nature of this element, only the matrix columns corresponding to the adders will contain two entries. The columns corresponding to the other elements will have just one entry.

Table 1. Adjacency matrix representation for the three basic elements

	Schematic symbol	Oriented graph	Adjacency matrix
Adder			
Multiplier			
Delay			

As an example, the proposed adjacency matrix representation of the first-order direct-form digital filter realization shown in Fig. 2(a) is given in Fig. 2(d).

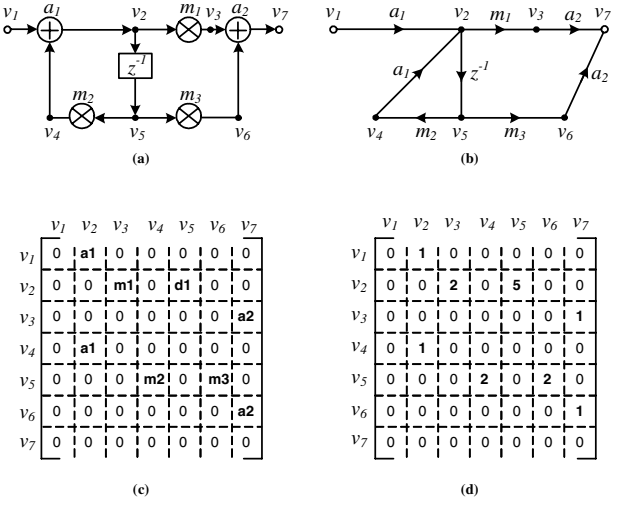


Fig. 2. Adjacency matrix representation for a first-order digital filter. (a) digital filter structure (b) oriented graph representation (c) adjacency matrix representation using acronyms (d) adjacency matrix representation replacing the acronyms by numbers.

In Fig. 2(a) the acronyms a, m and d represent adders, multipliers and delays, respectively. In Fig. 2(b) the digital filter is represented by its oriented graph. In Fig. 2(c) the adjacency matrix representation using the acronyms is depicted. Finally, in Fig. 2(d) the acronyms are replaced by a choice of numbers that will be explained in the next section.

As a rule, the input and the output nodes of all digital filters are, respectively, the lowest and the highest numbered nodes of the graph. Accordingly, the first column and the last line of the adjacency matrix are empty since, as shown in Fig. 2(a), there is no edges incident into the input or incident out of the output nodes.

3 Invalid Individuals

The efficiency of an evolutionary algorithm requires suitable chromosome coding schemes to detect anomalous individuals generated during the evolutionary process [7]. Invalid individuals can be generated in the initial population or after the application of the genetic operators. In the case of digital filters, there are two possible anomalies: structurally invalid individuals and non-computable individuals containing delay-free loops [8]. The first category consists of filters that have more than one or no component at all incident into a circuit node. The second category is composed of physically unrealizable filters that would demand digital processors with infinite bandwidth to operate properly.

3.1 Structurally Invalid Individuals

Since in digital filters the circuit components depend on the signal flow direction, topological restrictions are particularly severe when compared with analog filters [5]. The structural anomalies that may occur in a digital filter are listed in Table 2 with their corresponding adjacency matrix representation. In this table the first structure has a delay and a multiplier incident into the same node v_k as indicated by the two entries in the corresponding column of the matrix. The second anomalous structure is the short-circuiting self-loop in some node v_j which is represented by a nonzero entry in the corresponding entry of the matrix main diagonal. Finally, the third anomalous structure is that containing an isolated node, such as node v_j in the third row of Table 2. This corresponds to an empty line in the adjacency matrix.

Therefore, in order to detect a structurally invalid individual in the initial population and during the execution of the evolutionary synthesis it is sufficient to apply the following set of rules:

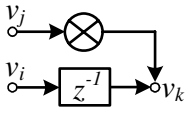
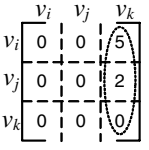
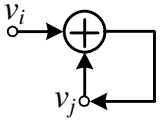
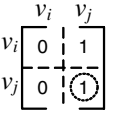
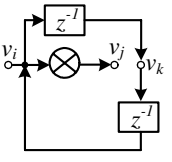
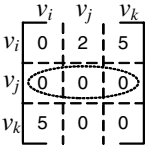
Rule 1: Each column of the adjacency matrix, except the first one, must contain only one type of element. To implement this rule, the numbers 1, 2 and 5 are assigned to the three basic elements of a digital filter. With this particular choice of numbers, the rule is easily verified by summing the entries of each column of the matrix: the columns containing an adder or a multiplier must sum 2 and those containing a delay must sum 5. Thus, a structurally invalid individual is created if the sum of any column is different from 2 or 5.

Rule 2: Since the flow graphs of digital filters have no ground node, the main diagonal entries of the adjacency matrix must be zero.

Rule 3: Except for the last line corresponding to the output node, the adjacency matrix can not contain empty lines. Thus, each ordinary vertex of the graph must have at least one edge incident out of it.

The last necessary step to validate an individual is to verify its computability.

Table 2. Structurally invalid structures

Rule	Invalid Structure	Adjacency Matrix
1		
2		
3		

3.2 Computable Digital Filters

A digital filter is said to be computable if it does not contain delay-free loops. An example of a structure containing a delay-free loop is given in Fig. 3.

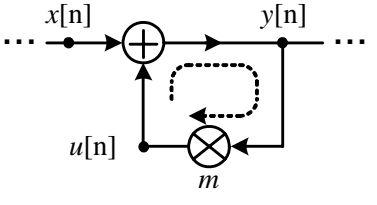


Fig. 3. Example of a digital filter containing a delay-free loop

The system of equations derived from this structure is:

$$\begin{cases} y[n] = x[n] + u[n] \\ u[n] = m \cdot y[n] \end{cases} \tag{2}$$

The combination of these two equations will result in a non-computable filter as given by:

$$y[n] = x[n] + m \cdot y[n] \tag{3}$$

where, the output current value $y[n]$, depends on itself to be evaluated.

This is physically impossible due to the fact that any digital processor requires a finite time to perform arithmetic operations.

To test for the computability of an individual with the proposed chromosome coding scheme it is necessary to introduce the path matrix:

Let $G(V,E)$ be an oriented graph with n vertices, $v_1, v_2, ..., v_n$. Assuming the usual definition of an oriented path in an oriented graph [6], the path matrix of G is a $n \; x \; n$ matrix $P = [p_{ij}]$ defined by:

$$p_{ij} = \begin{cases} 1, & \text{if there is a directed path from } \mathrm{v_i} \text{ to } \mathrm{v_j} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The path matrix is obtained from the adjacency matrix through Warshall's algorithm [9]. This algorithm determines all possible paths between every pair of vertices in the graph with a complexity $O(n^3)$. Considering the overall cost of the evolutionary algorithm and the relatively small number of nodes used in digital filters synthesis, the cost of Warshall's algorithm does not represent an inconvenient.

As an example, the path matrix P corresponding to the graph of Fig. 4(a) is shown in Fig. 4(b). It can be seen in the figure that, if $i \neq j$ and $p_{ij} = 1$ there is a path between vertices v_i and v_j, otherwise, if $i = j$, there is a loop. Therefore, a non-zero entry in the path matrix main diagonal indicates a loop in the filter structure.

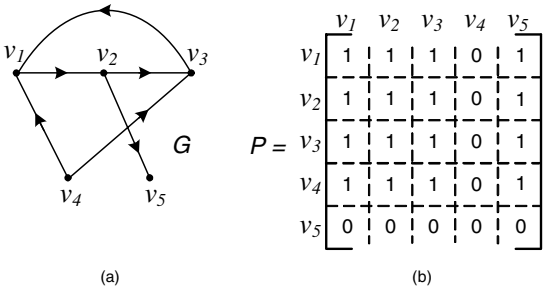


Fig. 4. (a) Path matrix (b) corresponding oriented graph G

Thus, using the adjacency matrix representation, the computability of digital filters is verified by the following algorithm:

Computability Verification Algorithm

S1. Generate the adjacency matrix following the rules of Section 3.1;

S2. Replace the delays by zeros in the matrix;

S3. Obtain the path matrix P using the Warshall's algorithm;

S4. If all main diagonal entries of matrix P are zeros the filter is computable, otherwise it is non-computable and the individual is not valid.

An example of a non-computable filter with a delay-free loop (indicated by a dotted line) is given in Fig. 5(a). The delay-free loop extends through nodes 2, 4 and 5 as indicated by the main diagonal entries of the path matrix in Fig. 5(d).

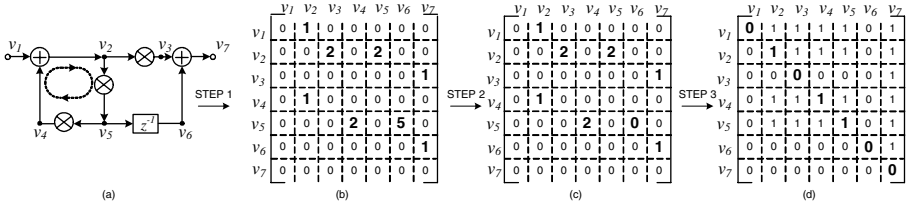


Fig. 5. Verification of the computability for a first-order filter (a) digital filter structure (b) corresponding adjacency matrix representation, (c) delay-free adjacency matrices and (d) path matrices

4 Genetic Operators

The crossover strategy used in the proposed evolutionary synthesis method is illustrated in Fig. 6. It consists in exchanging two submatrices with randomly chosen dimensions. Assuming two adjacency matrices of dimensions m and n with $m < n$, the coordinates of the crossover point (i, j) are chosen on the smaller matrix. The integers i and j are selected inside the intervals:

$$i \in [0,\ m-1] \tag{5a}$$

$$j \in [0,\ m-1] \tag{5b}$$

In order to define the dimensions of the submatrices to be exchanged, two integers p and q are randomly chosen in the intervals:

$$p \in [1,\ m-i] \tag{6a}$$

$$q \in [1,\ m-j] \tag{6b}$$

Finally, the coordinates of the crossover point on the largest matrix are integers randomly chosen in the intervals:

$$k \in [0,\ n-p] \tag{7a}$$

$$l \in [0,\ n-q] \tag{7b}$$

An example of crossover between two simple first-order filters is shown in Fig. 6. In this figure, the dimensions of the matrices, the coordinates of the crossover points and the dimensions of the submatrices are, respectively, $m = n = 4$, $i = j = k = l = 1$, $p = 2$ and $q = 3$. It is easy to verify that the resulting individual in Fig. 6(d) is structurally invalid: the sum of the entries in the columns 2 and 3 of the adjacency matrix violates Rule 1 in Section 3.1.

The mutation operator simply replaces a randomly chosen matrix column by a new one filled with an also randomly chosen element. Fig. 7 shows an example of the proposed mutation scheme. In this figure, the delay that was incident out of v_3 before the mutation becomes, after the mutation, a multiplier incident out of v_5.

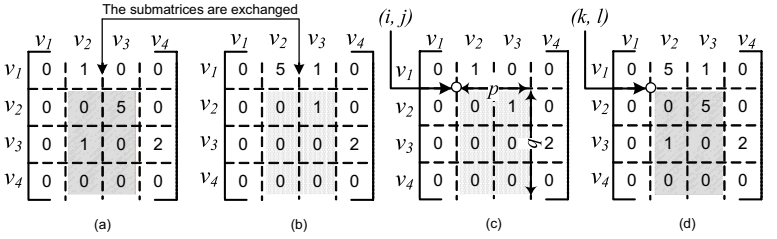


Fig. 6. Example of a crossover between two first-order digital filters. (a) parent 1 (b) parent 2 (c) offspring individual 1 (d) offspring individual 2.

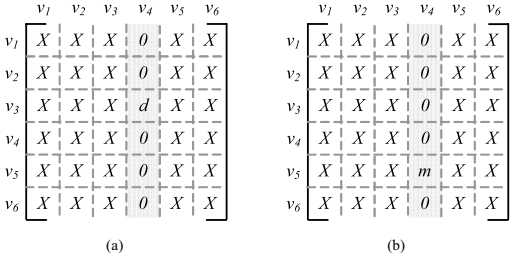


Fig. 7. Adjacency matrix representation of an individual before (a) and after (b) mutation

5 Comparison with the GP Approach

Evolutionary methods to synthesize low-sensitivity digital filters, as those proposed by Uesaka and Kawamata [10-11], are generally based on the GP-tree coding scheme with automatically defined functions (ADFs) [2].

An example of a chromosome obtained by this approach is shown in Fig. 8. The filter structure is described by a program in the Lisp language from which the rooted-tree depicted in Fig. 8(b) is constructed. In this figure, the nodes above the dotted line are divided in two categories: function-defining branches and result-producing branch. The first category is composed of nodes v_0 up to v_7, while the second category is composed of the output node y. The subtrees below each node are named bodies. The bodies, in turn, comprise a set of functions and arguments. Therefore, the function-defining branch corresponding to node v_5 consists of the set of functions $\{a_1, a_3, a_4, m_2\}$ and the set of arguments $\{v_1, v_2, v_3\}$. The nodes at the origin of the function-defining branches and the result-producing branch are fixed and are not affected by the genetic operators.

5.1 GP Crossover Operator

In order to compare the crossover operation in the GP-tree and in the adjacency matrix approaches, consider the function-defining branch corresponding to node v_6 in Fig. 8(b). In the GP-tree approach any node in the bodies can be selected as the crossover point. The six possible crossover points in this case are numbered and indicated by dotted circles in Fig. 9(a).

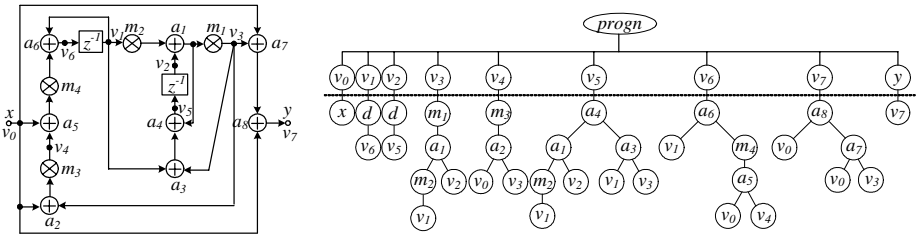


Fig. 8. (a) Structure of the second-order filter (b) rooted GP tree

To obtain the adjacency matrix representation corresponding to this branch, the tree structure must be translated into an oriented graph as shown in Fig. 9(b). The graph is generated by mapping the GP-tree functions and arguments into the graph edges and vertices respectively. Note that in order to perform the mapping it is necessary to define two additional arguments v_x and v_y in the GP-tree, since there are no incident arguments into function node m_4.

In Fig. 9(c), two additional crossover points that are not possible in the GP-tree are shown. The corresponding adjacency matrix representation is given in Fig. 9(f).

For the sake of clarity, the submatrices of the adjacency matrix corresponding to the possible crossover points of Fig. 9(a) are represented in Figs. 9(d) and 9(e). Thus, all possible crossover points in the GP-tree can be mapped in the proposed representation by the crossover scheme described in the previous section. For example, the crossover point (3) in Figs. 9(a) and 9(b) is defined by the following parameters in the proposed crossover scheme: $i = 0$, $j = 4$, $p = 1$ and $q = 3$.

5.2 Mutation

In the GP approach, a mutation creates a new function-defining branch by replacing an existing argument by another possible argument [10]. As an example take the same function-defining branch v_6 discussed previously. The argument symbols that can be replaced by the mutation operator are indicated by dotted ellipses numbered from (1) to (3) in Fig. 10(a).

In the adjacency matrix representation, replacing an argument symbol is equivalent to move the corresponding function entry in the matrix along the entries of a given column, as indicated by the dotted arrows in Fig. 10(b). For example, if the argument v_4 indicated by ellipse (1) in Fig. 10(a) is replaced during a mutation operation by the argument v_1 as shown in Fig. 10(c), this will correspond to move the function a_5 from the position (row v_4, column v_x) indicated in Fig. 10(b) to the position (row v_1, column v_x) in Fig. 10(d).

Note that the GP mutation scheme can only change an argument symbol by another argument symbol, maintaining the associated function symbols unchanged. In contrast, the mutation scheme proposed in Section 4 may operate in any element of the

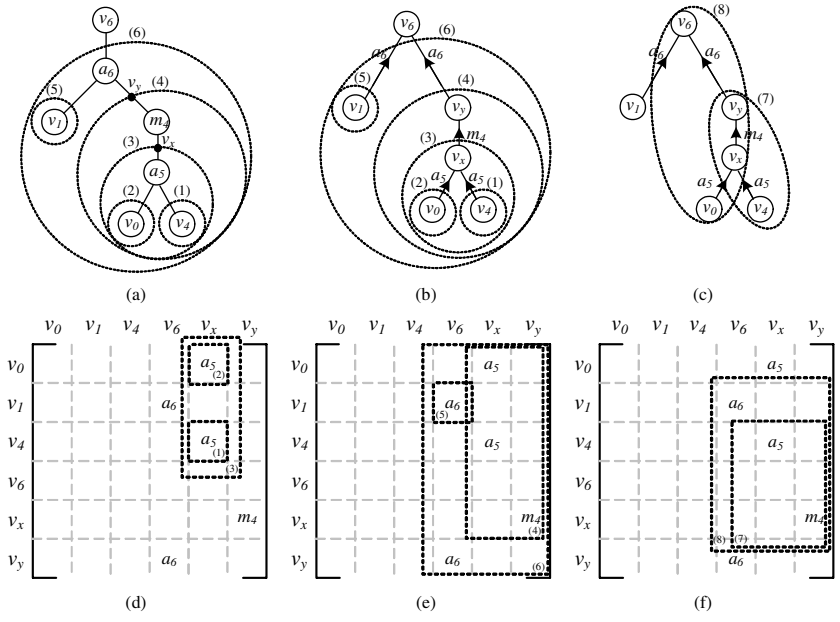


Fig. 9. (a) GP tree of the function-defining branch corresponding to node v_6 of Fig. 8(b) (b) graph associated with the adjacency matrix representation indicating the same crossover points of the GP-tree representation (c) graph associated with the adjacency matrix representation indicating two crossover points that are not possible in the GP-tree (d) adjacency matrix corresponding to the GP-tree indicating the crossover points (1), (2) and (3) (e) adjacency matrix corresponding to the GP-tree indicating the crossover points (4), (5) and (6) (f) adjacency matrix corresponding to the two additional crossover points (7) and (8)

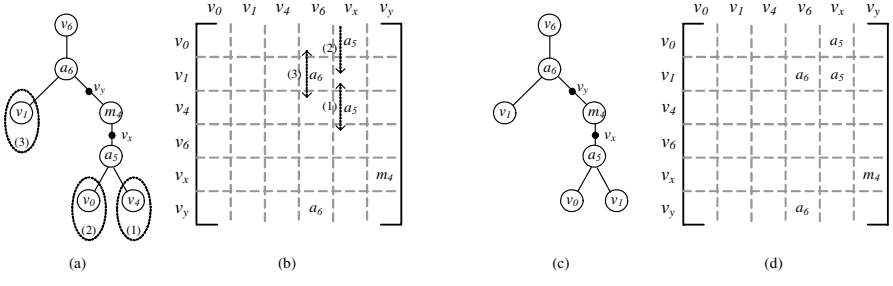


Fig. 10. (a) GP tree and (b) adjacency matrix before the mutation (c) GP tree and (d) adjacency matrix after the mutation

tree by moving the matrix entries along the columns eventually changing the type of the function symbol.

Therefore, the proposed representation is associated with larger search spaces than those possible with the GP-tree representation.

6 Numerical Results

In order to compare the performance of the proposed evolutionary algorithm with other approaches found in the literature, the following low-pass second-order IIR digital filter is used [11]:

$$H(z) = \frac{0.098244z^2 - 0.195065z + 0.098244}{z^2 - 1.957184z + 0.9586936} \tag{8}$$

This filter has poles close to the unit circle being particularly sensitive to the effect of the coefficients quantization errors which in the worst case can cause instability. The design of this class of filters is not a straightforward task since there are no general analytical methods to derive the optimal filter structure.

The multipliers values are constrained to the interval [-1, +1] in order to increase the number of bits representing their fractional part after the quantization process. Table 3 shows the parameters of the evolutionary process used in the example.

Table 3. Parameters of the evolutionary process

Population size	200	Multipliers interval	[-1,+1]
Number of generations	50	Maximum number of multipliers	4
Crossover rate	70%	Maximum number of nodes	21
Mutation rate	20%	E_{sens}	100

The fitness computation is essential in any evolutionary process to verify if an individual realizes a given target transfer function. In the proposed example, the fitness function is computed as the inverse of the function:

$$\varepsilon = 1 + \varepsilon_{sens} \tag{9}$$

Where ε_{sens} is given by:

$$\varepsilon_{sens} = \begin{cases} S, \text{ if } \mathrm{H}_{\mathrm{target}}(\mathrm{z}) \text{ is realized} \\ E_{sens}, \text{ otherwise} \end{cases} \tag{10}$$

In these equations, E_{sens} is a constant and S is the sensitivity of the digital filter defined as [12]:

$$S = \sum_{i=1}^{N} \max_{\omega} \frac{\partial \left| H(e^{j\omega}) \right|}{\partial m_i} \tag{11}$$

Where ω is the frequency, N is the number of multipliers, $\left| H\left(e^{j\omega}\right) \right|$ is the transfer function magnitude and m_i is the ith multiplier coefficient. The choice of the constant E_{sens} is based on many runs of the evolutionary algorithm until reasonable values of the fitness are achieved. Generally some kind of heuristics based on expert knowledge of the design problem is required [13].

Fig. 11(a) shows the average fitness and the best fitness throughout the generations. As shown in the figure, the algorithm responds well to genetic operators,

increasing considerably the fitness along the generations. This suggests that the checking rules included to avoid the evaluation of invalid individuals do not constrain the search space and do not reduce significantly the overall algorithm performance. The synthesized topology after 50 generations is shown in Fig. 11(b). The synthesized structure is said to be canonic, since the number of delays is equal to the order of the transfer function. Moreover, it uses a minimal number of multipliers that represents an important feature since the multipliers are the most expensive components in any digital filter implementation.

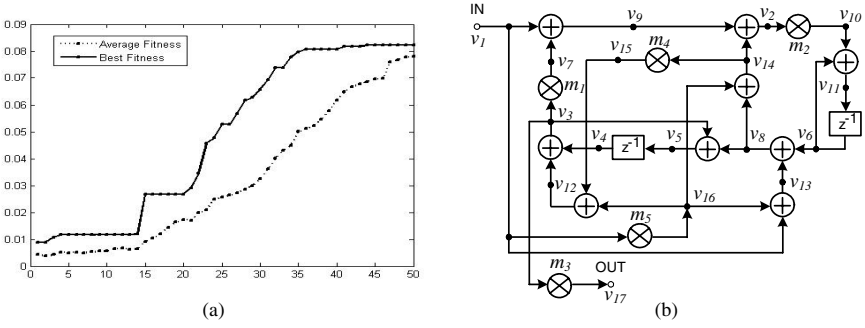


Fig. 11. (a) The average fitness and the best fitness of the proposed evolutionary algorithm along the generations for the transfer function (8) (b) synthesized topology

A comparison of the results obtained by different methodologies found in the literature to synthesize low-sensitivity structures is given in Table 4. All results refer to the second-order transfer function (8). The first three lines of the table use conventional approaches, whereas the last two lines use evolutionary algorithms. It can be observed that the smallest magnitude sensitivity was obtained with the topology synthesized by the proposed evolutionary algorithm.

Table 4. Sensitivity of the transfer function (8)

Direct Form II	2393.883
Diniz & Antoniou [14]	1317.327
Balanced form [15]	75.817
Uesaka & Kawamata [11]	15.370
Synthesized topology	11.182

7 Conclusions

A chromosome coding scheme based on the adjacency matrix that increases the search space dimension by report to GP-tree schemes is proposed to perform the evolutionary synthesis of low-sensitivity digital filters.

It is shown that applying simple rules requiring minimal computational effort, the proposed coding scheme representation has the property of detecting the generation of

structurally invalid individuals and non-computable ones during the evolutionary process.

It was observed that the results obtained by the proposed evolutionary method using the adjacency matrix representation compares favorably with other results found in the literature.

References

1. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
2. Koza, J.R., et al.: Use of Automatically Defined Functions and Architecture-Altering Operations in Automated Circuit Synthesis with Genetic Programming. In: Proc. of the First Annual Conference (1996)
3. Poli, R.: Evolution of graph-like programs with parallel distributed genetic programming. In: Proc. of the 7th International Conference on Genetic Algorithms, pp. 346–353 (1997)
4. Miller, J.F., Thomson, P.: Cartesian Genetic Programming. In: Poli, R., Banzhaf, W., Langdon, W.B., Miller, J., Nordin, P., Fogarty, T.C. (eds.) EuroGP 2000. LNCS, vol. 1802, pp. 121–132. Springer, Heidelberg (2000)
5. Mesquita, A., Salazar, F.A., Canazio, P.P.: Chromosome Representation through Adjacency Matrix in Evolutionary Circuits Synthesis. In: Proc. of the 2002 NASA/DoD Conference on Evolvable Hardware, pp. 102–109. IEEE Computer Press, Alexandria (2002)
6. Swamy, M.N.S., Thulasiraman, K.: Graphs, Networks and Algorithms. John Wiley & Sons, Chichester (1981)
7. Sapargaliyev, Y., Kalganova, T.: Constrained and Unconstrained evolution of LCR low-pass filters with oscillating length representation. IEEE Congress on Evol. Comput., 1529–1536 (2006)
8. Oppenheim, A.V., Schafer, R.W.: Digital Signal Processing. Prentice-Hall International, London (1975)
9. Warshall, S.: A Theorem on Boolean Matrices. Journal ACM 9, 11–12 (1962)
10. Uesaka, K., Kawamata, M.: Evolutionary Synthesis of Digital Filter Structures Using Genetic Programming. IEEE Trans. Cir. and Syst. II 50, 977–983 (2003)
11. Uesaka, K., Kawamata, M.: Synthesis of low-sensitivity second-order digital filters using genetic programming with automatically defined functions. IEEE Signal Processing Lett. 7, 83–85 (2000)
12. Crochiere, R.E., Oppenheim, A.V.: Analysis of linear digital networks. In: Proc. IEEE, vol. 63, pp. 581–595 (1975)
13. Zebulum, R.S., Pacheco, M.A.C., Vellasco, M.M.B.R.: Evolutionary Electronics - Automatic Design of Electronic Circuits and Systems by Genetic Algorithms. CRC Press, Boca Raton (2001)
14. Diniz, P.S.R., Antoniou, A.: Low-sensitivity digital-filter structures which are amenable to error spectrum shaping. IEEE Trans. Circuits Syst. 32, 1000–1007 (1985)
15. Kawamata, M., Iwatsuki, M., Higuchi, T.: Balanced realizations as minimum sensitivity structures in linear systems. Trans. Soc. Instrum. Contr. Eng. 21, 900–906 (1985)

Using Hyper-heuristics for the Dynamic Variable Ordering in Binary Constraint Satisfaction Problems

Hugo Terashima-Marín[1], José C. Ortiz-Bayliss[1], Peter Ross[2], and Manuel Valenzuela-Rendón[1]

[1] Tecnológico de Monterrey - Center for Intelligent Systems, Monterrey, NL, 64849, Mexico
{terashima,a00796625,valenzuela}@itesm.mx
[2] School of Computing, Napier University, Edinburgh EH10 5DT UK
P.Ross@napier.ac.uk

Abstract. The idea behind hyper-heuristics is to discover some combination of straightforward heuristics to solve a wide range of problems. To be worthwhile, such combination should outperform the single heuristics. This paper presents a GA-based method that produces general hyper-heuristics for the dynamic variable ordering within Constraint Satisfaction Problems. The GA uses a variable-length representation, which evolves combinations of condition-action rules producing hyper-heuristics after going through a learning process which includes training and testing phases. Such hyper-heuristics, when tested with a large set of benchmark problems, produce encouraging results for most of the cases. There are instances of CSP that are harder to be solved than others, this due to the constraint and the conflict density [4]. The testebed is composed of hard problems randomly generated by an algorithm proposed by Prosser [18].

1 Introduction

A Constraint Satisfaction Problem (CSP) is defined by a set of variables X_1, X_2, ..., X_n and a set of constraints C_1, C_2, ..., C_m. Each variable X_i has a nonempty domain D_i of possible values. Each constraint C_i involves some subset of variables and specifies the allowable combinations of values for that subset [21]. When trying to find a solution for this kind of problems, it is impossible to avoid the implicit issue of determining which variable is the next to be instantiated. Many researchers have proved, based on analysis and experimentation, the importance of the variable ordering and its impact in the cost of the solution search [18]. This ordering has repercussions in the complexity of the search, which means that finding methods that help to efficiently order these variables is an important issue. For small combinatorial problems, exact methods can be applied. However, when larger and more complex problems appear, exact solutions are not a reasonable choice since the search space grows exponentially, and so does the time for finding the optimal order. Various heuristic

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 407–417, 2008.

and approximate approaches have been proposed that guarantee finding near optimal solutions. However, it has not been possible to find a reliable method that is able to solve all instances of a given problem. In general, some methods work well for particular instances, but not for all of them.

It is possible to establish different criteria to face the ordering problem, and it can be done either in static or in dynamic fashion. In the static way, the order of the variables is set from the start and is kept during the complete search procedure. This ordering does not guarantee to find a feasible solution given that it is a permutation problem, deriving in an exponential growth in the number of variables. In the other hand, in the dynamic variable ordering, the order is constructed during the search, based on some criterion about the characteristics of the variables left to instantiate. With the dynamic ordering the search space is not as large as in the static ordering. It has been proved in many studies that dynamic ordering gives better results than static ordering [6]. When working with dynamic ordering we can use heuristics to select the next variable to instantiate.

The aim of this paper is to explore a novel alternative on the usage of evolutionary approaches to generate hyper-heuristics for the dynamic variable ordering in CSP. A hyper-heuristic is used to define a high-level heuristic that controls low-level heuristics [3]. The hyper-heuristic should decide when and where to apply each single low-level heuristic, depending on the given problem state. The choice of low-level heuristics may depend on the features of the problem state, such as CPU time, expected number of solutions, values on the objective function, etcetera. Selecting a particular heuristic is dynamic, and depends on both the problem state produced by the previous heuristic applied and the search space to be explored in that point of time.

The investigation in this article, presents a method to generate a general hyper-heuristic intended to provide a way to order variables for a wide variety of hard instances of CSP, so they can be solved. The procedure learns the hyper-heuristic by going through a training phase using hard instances with different features. The generated hyper-heuristic is tested later with a collection of unseen examples providing acceptable results. The general method is based on a variable-length Genetic Algorithm, where the chromosome is conformed of a series of blocks, representing condition-action rules.

The paper is organized as follows. Section 2 presents the variable ordering problem, the solution method proposed and its justification. This is followed by the experimental setup, the results, their analysis and discussion in section 3. Finally, in section 4 we include our conclusions and some ideas for future work.

2 Solution Approach

In the literature, one can see that Evolutionary Computation has been used in few CSP investigations [5,15]. Recently, Terashima et al used a combination between low level heuristics and a Genetic Algorithm for dynamic variable ordering in CSP's but did not incorporate the concept of hyper-heuristic [23]. More re-

cently, Terashima et al also used a GA based method to produce hyper-heuristics for the 2-D cutting stock problem with outstanding results [24]. Evolutionary Computation usually includes several types of evolutionary algorithms [25]: Genetic Algorithms [14,10], Evolutionary Strategies [19,22], and Evolutionary Programming [7,1]. In this research we use a GA with variable length chromosomes, a resemblance of what is called a *messy*-GA [11].

2.1 The Variable Ordering Problem in CSP

In CSP, the variable ordering is relevant topic due to its impact in the cost of the solution search. As we mentioned before, there are two different ways to set this order: static and dynamic. This research focuses on the dynamic fashion, which uses heuristics to select the next variable. It has been empirically proved in previous studies that dynamic variable ordering is more efficient than the static approach [6]. Also, the literature is rich in heuristics designed for this task: Fail-First [13], Saturation Degree [2], *Rho* [9], *Kappa* [9] and *E(N)* [9], to mention some. However, none of these heuristics has been proved to be efficient in every instance of the problems.

Every time an instantiated variable is checked to verify if it does not violate any constraint is called a constraint check. Counting the constraint checks used to solve a specific instance is used in this work to compare the performance of the different heuristics and the general hyper-heuristic.

Some CSP exhibit a phase transition, where problems change from having a solution to not having any, and this occurs at a critical value of connectivity. Problems that occur below the critical value of connectivity are easy to solve, and problems above the critical value have not a solution because of the high value of connectivity. It is only around the critical value of connectivity that is hard to decide if a problem has a solution [18] and these instances requiere a great computational effort [4]. In this research, we used only hard instances of CSP.

2.2 The Set of Heuristics Used

In CSP, the related heuristics refer to the way the next variable is selected and which value has to be used to instantiate that variable. In this investigation we focus on the first kind of heuristics, those related to the order in which variables are selected to be instantiated. Value ordering is beyond the scope of this research and it is left for future work. Some of the heuristics were taken from the literature and others were adapted. We chose the most representative heuristics in its type, considering their individual performance presented in related studies and also in an initial experimentation on a collection of benchmark problems.

The ordering heuristics used in this research are: Fail First (FF), Modified Fail First (MFF), Saturation Degree (Bz), *Rho (R)*, E(N), *Kappa (K)*, and Min-Conflicts (MC). These heuristics, except for MFF and MC are described in detail by Gent et al. [9]. MFF is a modification of the Fail-First heuristic and the MC heuristic is a very simple heuristic based on the idea of selecting a variable that produces the subproblem that minimizes the number of conflicts in the variables

left to instantiate. This heuristic was originally used for value ordering [17,16], however it was adapted to work for variable ordering.

2.3 Combining Heuristics with the Proposed GA

The concept of hyper-heuristic is motivated by the objective to provide a more general procedure for optimization [3]. Meta-heuristics methods usually solve problems by operating directly on the problem. Hyper-heuristics deal with the process to choose the right heuristic for solving the problem at hand. The aim is to discover a combination of simple heuristics that can perform well on a whole range of problems. For real applications, exhaustive methods are not a practical approach. The search space might be too large, or the number and types of constraints may generate a complex space of feasible solutions.

It is common to sacrifice quality of solutions by using quick and simple heuristics to solve problems. Many heuristics have been developed for specific problems. But, is there a single heuristic for a problem that solves all instances well? The immediate answer is no. Certain problems may contain features that would make a specific heuristic to work well, but those features may not be suitable for other heuristics. The idea with hyper-heuristics is to combine heuristics in such a way that a heuristic's strengths make up for the drawbacks of another.

The solution model used in this investigation carries features from previous work by Ross et al. [20] and Terashima et al. [24], in which the main focus is to solve one dimensional and two dimensional bin-packing problems, respectively. In the research presented in this article, a GA with variable-length individuals is proposed to find a combination of single heuristics to order variables to solve efficiently a wide variety of instances of CSP.

The basic concept is that, given a problem state P, this is associated with the closest point in the chromosome which carries the ordering rules to be applied. This application will transform the problem to a new state P'. The purpose is to solve a problem by constructing the answer, deciding on the heuristic to apply at each step.

A chromosome in the messy GA represents a set of labelled points within this simplified problem state space; the label of any point is a heuristic. The chromosome therefore represents a complete recipe for solving a problem, using a simple algorithm: until the problem is solved, (a) determine the current problem state P, (b) find the nearest point to it, (c) apply the heuristic attached to the point, and (d) update the state. The GA is looking for the chromosome (representing a hyper-heuristic) which contains the rules that apply best to any intermediate state in the solving process of a given instance.

The instances are divided into three groups: the training and the testing set I and II. The general procedure consists in solving first all instances in the three sets with the single heuristics. This is carried out to keep the best solution that is later used also by the GA we propose. The next step is to let the GA work on the training set until termination criterion is met and a general hyper-heuristic is produced. All instances in the testing and training sets I and II are then solved with this general hyper-heuristic.

Representation. Each chromosome is composed of a series of *blocks*. Each block j includes nine numbers. The first eight represent an instance of the problem state. The first number indicates the percentage of variables that remain to be instantiated (v_j). The second number represents the percentage of values in all the domains left (d_j). The next four numbers are related to the constraints that still remain to be instantiated. For this, we defined a three category classification for constraints: large constraints are those that prohibit a fraction $p_c \geq 0.55$, medium constraints have a $0.45 \leq p_c < 0.55$ and small constraints have a $p_c < 0.45$. These values were obtained through previous experimentation. With these categories we defined that the third number represents the percentage of all constraints(c_j), the fourth number indicates the percentaje of large constraints (lc_j), the fifth one shows the percentage of medium constraints (mc_j), and the sixth number represents the percentage of small constraints (sc_j). The next two numbers represent the value of the *rho* (r_j) and *kappa* (k_j) factors, respectively. The last number represents the variable ordering heuristic associated to the instance of the problem state. For a given problem state, the inital eight numbers would lie in a range between 0 and 1, so that the actual problem state is a point inside the unit eight-dimensional space. Nevertheless, we allow the points defined in each block to lie outside the unit cube, so we redefined the range to be from -3 to 3. At each step, the algorithm applies the heuristic that is associated to the block that is closest to actual problem state. We measure the distance d between the problem state P' and the instance inside each block j using the euclidean distance formula.

The Fitness Function. The most common criterion to measure the efficiency of a search algorithm used to determine if a CSP instance has or does not have a solution is the count of the consistency checks made during the search. To calculate the quality of an individual there are two steps that must be done:

1. Every CSP instance is solved using each low level heuristic. The CSP instance solver combines backtracking and forward checking to solve the instances. The best heuristic result, for each specified instance i is stored (let us call it BH_i). These results are prepared in advance of running the GA.
2. Each one of the problems assigned to the individual is solved using the coded hyper-heuristic. The best result of the low level heuristics (BH_i) is divided by the number of consistency checks used during the search (HH_i). Combining all together, the fitness function (FF_i) is computed as: $FF = \frac{BH_i}{HH_i}$. This evaluation guarantees that, when the general hyper-heuristic produces better results, the fitness function will return a number greater than one.

To compute the fitness for each chromosome, the distance between the solution obtained by that individual with respect to the best result given by the single heuristic (BH_i) is measured. The fitness is an average, and it is given by: $FF_m^l = \frac{FF_m^{l-1} \cdot mp_m + \sum_{i=1}^{5} FF(m_i)}{mp+5}$, where FF_m^{l-1} is the fitness for individual m in the previous generation; mp_m is the number of problems individual m has seen so far; $FF(m_i)$ is the fitness obtained by individual m for the each problem assigned to the hyper-heuristic. After each generation l, a new problem is assigned

to each individual m in the population and its fitness is recomputed again by: $FF_m^l = \frac{FF_m^{l-1} \cdot mp_m + FF(m)}{mp+1}$.

In both cases the evaluation is the result of both the past fitness and the current fitness. A very good performance for certain instances is not very significant if for most of the instances an individual has a low performance. In the other hand, if a hyper-heuristic behaves really good for most of the problems, a bad result for a single case does not affect negatively too much.

3 Experiments and Results

This section presents the experiments carried out during the investigation and the results obtained. The benchmark set is composed of randomly generated hard instances using the algorithm proposed by Prosser [18]. This algorithm generates instances of CSP characterized by the 4-tuple $< n, m, p_1, p_2 >$, where n is the number of variables, m is the uniform domain size, p_1 is the probability that a constraint exists between a pair of variables, and p_2 as the tightness of constraints [18]. The collection includes 1250 different instances: 600 for the training set, 400 for the testing set I and 250 for the training set II. Both the training and the testing set I are composed by distinct instances with $n = 20$ and $m = \{10, 20\}$ (A and B instances). The training set II contains instances with $n = 30$ and $m = 10$ (C instances). A more detailed description of the instances and the number of instances generated for every set is presented in Table 1.

Table 1. Description of the hard instances

Class	n	m	p_1	p_2	Training set	Testing set I	Training set II
A1	20	10	0.20	0.65	60	40	0
A2	20	10	0.40	0.45	60	40	0
A3	20	10	0.60	0.33	60	40	0
A4	20	10	0.80	0.27	60	40	0
A5	20	10	1.00	0.22	60	40	0
B1	20	20	0.10	0.90	60	40	0
B2	20	20	0.20	0.75	60	40	0
B3	20	20	0.30	0.63	60	40	0
B4	20	20	0.40	0.55	60	40	0
B5	20	20	0.50	0.48	60	40	0
C1	30	10	0.10	0.75	0	0	50
C2	30	10	0.20	0.53	0	0	50
C3	30	10	0.30	0.40	0	0	50
C4	30	10	0.40	0.30	0	0	50
C5	30	10	0.50	0.27	0	0	50

The training set was used to generate the general hyper-heuristic (HH) shown in Table 2, which represents the best individual in the last generation of the training process from the best result of ten runs. For each run, the parameters used to generate the general HH were: a population size of 20 during 100 cycles, crossover probability of 1.0, and mutation probability of 0.1. The general HH obtained includes 4 rules indicating the different problem states and the associated heuristic to be applied.

Aiming at testing the model effectiveness, four experiments were carried out.

Table 2. General hyper-heuristic produced by the GA

v	d	c	lc	mc	sc	r	k	h
-0.075	0.643	-1.045	0.513	0.536	0.8169	-0.194	-0.1739	MC
-0.238	1.117	-0.02	-1.1259	0.24	-1.085	0.516	-0.646	*Rho*
0.083	-0.871	1.041	-0.025	-2.0139	0.713	-0.197	-2.257	*Kappa*
0.24	0.662	-0.493	1.089	0.138	-1.165	0.786	-0.083	$E(N)$

3.1 Experiment Type I

In this experiment we test the general hyper-heuristic (HH) with instances already seen. We solved the instances in the training set using the general HH from Table 2. Results are compared against those generated by the best result from the low-level heuristics and the average of constraint checks done by the low-level heuristics. Table 3 sumarizes the results from this experiment and clasifies them into three main categories: (1) the general HH uses less consistency checks than the best heuristic, (2) the general HH uses the same amount of consistency checks (a difference of $\pm 3\%$ is considered as if both approaches use the same amount of consistency checks) and (3) the general HH uses more consistency checks that the best result from the low-level heuristics.

Table 3. Results of experiment type I

	HH vs Best result of the low-level heuristics					HH vs average result of the low-level heuristics				
	Reduction		Equal	Increment		Reduction		Equal	Increment	
Class	> 15%	15% to 3%	± 3%	3% to 15%	> 15%	> 15%	15% to 3%	± 3%	3% to 15%	> 15%
A1	0.00	0.00	41.67	13.33	45	98.33	1.67	0.00	0.00	0.00
A2	0.00	0.00	56.67	16.67	26.67	100.00	0.00	0.00	0.00	0.00
A3	0.00	0.00	88.33	5.0	6.67	100.00	0.00	0.00	0.00	0.00
A4	0.00	5.00	65.00	26.67	3.33	100.00	0.00	0.00	0.00	0.00
A5	0.00	8.33	18.33	61.67	11.67	100.00	0.00	0.00	0.00	0.00
B1	0.00	0.00	55.00	8.33	36.67	95.00	5.00	0.00	0.00	0.00
B2	0.00	0.00	56.67	10.00	33.33	100.00	0.00	0.00	0.00	0.00
B3	0.00	0.00	65.00	10.00	25.00	100.00	0.00	0.00	0.00	0.00
B4	0.00	0.00	95.00	3.33	1.67	100.00	0.00	0.00	0.00	0.00
B5	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00

As we can observe, the results show that the HH presents a competitive behavior for most of the classes. Some instances in classes A4 and A5 were solved using less consistency checks than the best result of the low-level heuristics, a fact that was not present in other classes. The best results of the HH against the best result from the low-level heuristics were obtained for $m = 20$, $n = 10$, $p_1 = 0.8$ and $p_2 = 0.27$, parameters that correspond to class A4. The results of the general HH versus the average constraint checks of the low-level heuristics suggest that the HH is a good choice to solve different instances of CSP. The general HH is almost always much better than the average, representing a reduction greater than 15% in almost every instance. The outstanding results obtained when comparing the HH against the average of constraint checks is due to the bad behavior of some heuristics for certain of these instances. As we mentioned before, for some instances a low-level heuristic has a very good performance, but for other it is simply very poor. These bad results raise the average and it makes possible that the HH has a better perfomance.

3.2 Experiment Type II

The second experiment uses the same general HH used in the experiment type I. This time, the unseen instances of the testing set I were solved using the HH, and the results were compared against the results of the best low-level heuristic and the average of constraints checks used by the single heuristics for each instance. The instances used in this experiment were not used during the training phase, but they were generated with similar parameters. The results of this experiment are shown in Table 4.

Table 4. Results of experiment type II

	HH vs Best result of the low-level heuristics					HH vs average result of the low-level heuristics				
	Reduction		Equal	Increment		Reduction		Equal	Increment	
Class	> 15%	15% to 3%	± 3%	3% to 15%	> 15%	> 15%	15% to 3%	± 3%	3% to 15%	> 15%
A1	0.00	0.00	45.00	12.50	42.50	100.00	0.00	0.00	0.00	0.00
A2	35.00	7.50	2.50	2.50	52.50	100.00	0.00	0.00	0.00	0.00
A3	0.00	2.50	72.50	15.00	10.00	100.00	0.00	0.00	0.00	0.00
A4	0.00	10.00	65.00	17.50	7.50	100.00	0.00	0.00	0.00	0.00
A5	0.00	12.50	27.50	35.00	25.00	100.00	0.00	0.00	0.00	0.00
B1	0.00	0.00	37.50	17.50	45.00	95.00	2.50	2.50	0.00	0.00
B2	0.00	0.00	65.00	12.50	22.50	100.00	0.00	0.00	0.00	0.00
B3	0.00	0.00	87.50	2.50	10.00	100.00	0.00	0.00	0.00	0.00
B4	0.00	0.00	87.50	10.00	2.50	100.00	0.00	0.00	0.00	0.00
B5	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00

In this experiment, the results for class A2 are outstanding. In 42.5% of the instances the HH overcomes the best result of the low-level heuristics. This result is interesting, because the same parameters were used to generate the training set and the results for class A2 were not as good as in that case. It is natural to think that the results for a specific class in the training set should be similar to those in the testing set I for that class too, because they were both generated with the same parameters. It seems that the random numbers generator had a great impact in the generation for some classes of instances, in this case, for class A2. For classes A3 to A5 the results are similar to those presented in the training set: the HH achieves a reduction in the number of constraint checks used by the best result of the low-level heuristics in some cases. If we compare the results of the HH against the average of constraints checks of the low-level heuristics we observe again that the HH is much better than the average. For every class of instances, the HH uses a very reduced number of constraints checks compared to the used by the average of the low-level heurisics.

3.3 Experiment Type III

For experiment type III we used the general HH presented in the last two experiments. In this experiment, the instances of the testing set II were solved using the HH, and the results were compared as before. The instances used in this experiment were not used during the training phase, and they were not generated with similar parameters. The results of this experiment are shown in Table 5.

Table 5. Results of experiment type III

	HH vs Best result of the low-level heuristics					HH vs average result of the low-level heuristics				
	Reduction		Equal	Increment		Reduction		Equal	Increment	
Class	> 15%	15% to 3%	± 3%	3% to 15%	> 15%	> 15%	15% to 3%	± 3%	3% to 15%	> 15%
C1	0.00	0.00	40.00	12.00	48.00	96.00	2.00	0.00	2.00	0.00
C2	0.00	0.00	68.00	10.00	22.00	100.00	0.00	0.00	0.00	0.00
C3	0.00	0.00	94.00	2.00	4.00	100.00	0.00	0.00	0.00	0.00
C4	0.00	0.00	60.00	2.00	38.00	84.00	10.00	6.00	0.00	0.00
C5	0.00	10.00	74.00	12.00	4.00	42.00	10.00	46.00	2.00	0.00

The instances used for this experiment were never seen by the HH and had different parameters to those used in training. In other words, the HH was not trained to solve these classes of instances. The results show that the HH is competitive, but it is unable to reduce the best result of the low-level heuristic for classes C1 to C4. For class C5 a reduction is present in 10% of the instances, but it is not a reduction greater that 15% in the number of constraints checks. When comparing the HH against the average of constraint checks used by the low-level heuristics, the results are as good as in the past three experiments. For class C5, the general HH was only able to achieve a reduction greater than 15% in 42% of the instances. The results are not as good as in the past two experiments, but they are still very promissing.

3.4 Experiment Type IV

For this experiment we compared the general Hyper-heuristic with each of the low-level heuristics when solving the testing set I. The performance of the low-level heuristics and the HH is shown in Table 6.

Table 6. Low-level heuristics and general hyper-heuristic compared against the best result of the low-level heuristics for the testing set I

	Reduction		Equal	Increment	
Method	> 15%	15% to 3%	± 3%	3% to 15%	> 15%
Kappa	0.00	0.00	73.00	10.00	17.00
HH	0.00	2.67	54.00	24.67	18.67
E(N)	0.00	0.00	31.00	28.67	40.33
FF	0.00	0.00	6.67	27.00	66.33
Rho	0.00	0.00	5.67	23.67	70.66
MC	0.00	0.00	2.67	0.33	97.00
MFF	0.00	0.00	1.33	0.67	98.00
Bz	0.00	0.00	0.00	18.33	81.67

As we can observe, the HH is competitive and is only overcome by the *kappa* heuristic, which shows the best results in 73% of the instances. The HH achieves a reduction in the minimum number of constraint checks required by the low-level heuristic for 2.67% of the instances, and uses the minimum number of constraint checks achieved by the heuristics for at least half of the instances.

4 Conclusions and Future Work

This document has described experimental results in a model based on a variable-length GA which evolves combinations of condition-action rules representing

problem states and associated selection heuristics for the dinamic variable ordering problem in CSP. These combinations are called hyper-heuristics. Overall, the scheme identifies efficiently general hyper-heuristics after going through a learning procedure with training and testing phases. When applied to unseen examples, those hyper-heuristics solve many of the problems efficiently, in some cases better than the best single heuristic for each instance. Ideas for future work involve extending the proposed strategy to solve problems including heuristics for value ordering and include other algorithms for the CSP solving module.

Acknowledgments

This research was supported in part by ITESM under the Research Chair CAT-010 and the CONACYT Project under grant 41515.

References

1. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic programming: An Introduction. Morgan Kaufmann Publishers, Inc., London (1998)
2. Brelaz, D.: New methods to colour the vertices of a graph. Comunications of the ACM 22 (1979)
3. Burke, E., Hart, E., Kendall, G., Newall, J., Ross, P., Schulenburg, S.: Hyper-heuristics: An emerging direction in modern research technolology. In: Handbook of Metaheuristics, pp. 457–474. Kluwer Academic Publishers, Dordrecht (2003)
4. Cheeseman, P., Kanefsky, B., Taylor, W.M.: Where the real hard problems are. In: Proceedings of IJCAI 1991, pp. 331–337 (1991)
5. Craenen, B.G.W., Eiben, A.E., van Hemert, J.I.: Comparing evolutionary algorithms on binary constraint satisfaction problems. IEEE Transactions on Evolutionary Computation 7(5), 424–444 (2003)
6. Dechter, R., Meiri, I.: Experimental evaluation of preprocessing algorithms for constraint satisfaction problems. Artificial Intelligence 38(2), 211–242 (1994)
7. Fogel, D.B., Owens, L.A., Walsh, M.: Artificial Intelligence through Simulated Evolution. Wiley, New York (1966)
8. Garey, M., Johnson, D.: Computers and Intractability. W.H. Freeman and Company, New York (1979)
9. Gent, I.P., MacIntyre, E., Prosser, P., Smith, B.M., Walsh, T.: An empirical study of dynamic variable ordering heuristics for the constraint satisfaction problem. In: Freuder, E.C. (ed.) CP 1996. LNCS, vol. 1118, pp. 179–193. Springer, Heidelberg (1996)
10. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading (1989)
11. Goldberg, D., Korb, B., Deb, K.: Messy genetic algorithms: Motivation, analysis and first results. Complex Systems, 93–130 (1989)
12. Golden, B.L.: Approaches to the cutting stock problem. AIIE Transactions 8, 256–274 (1976)
13. Haralick, R.M., Elliott, G.L.: Increasing tree search efficiency for constraint satisfaction problems. Artificial Intelligence 14, 263–313 (1980)
14. Holland, J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)

15. Marchiori, E., Steenbeek, A.: A genetic local search algorithm for random binary constraint satisfaction problems. In: Proceedings of the 2000 ACM symposium on Applied computing, Como, Italy, vol. 1, pp. 458–462 (2000)
16. Minton, S., Johnston, M.D., Phillips, A., Laird, P.: Minimizing conflicts: A heuristic repair method for csp and scheduling problems. Artificial Intellgence 58, 161–205 (1992)
17. Minton, S., Phillips, A., Laird, P.: Solving large-scale csp and scheduling problems using a heuristic repair method. In: Proceedings of the 8th AAAI Conference, pp. 17–24 (1990)
18. Prosser, P.: Binary constraint satisfaction problems: Some are harder than others. In: Proceedings of the European Conference in Artificial Intelligence, Amsterdam, Holland, pp. 95–99 (1994)
19. Rechenberg, I.: Evolutionstrategie: Optimierung technischer systeme nach prinzipien dier biolischen evolution. Frommann-Holzboog, Stuttgart (1973)
20. Ross, P., Blázquez, J.M., Schulenburg, S., Hart, E.: Learning a procedure that can solve hard bin-packing problems: A new ga-based approach to hyper-heuristics. In: Proceedings of GECCO 2003, pp. 1295–1306 (2003)
21. Russell, S., Norvig, P.: Artificial Intelligence A Modern Approach. Prentice Hall, Englewood Cliffs (1995)
22. Schwefel, H.P.: Numerical Optimization of Computer Models. Wiley, Chichester (1981)
23. Terashima-Marín, H., Calleja-Manzanedo, R., Valenzuela-Rendón, M.: Genetic Algorithms for Dynamic Variable Ordering in Constraint Satisfaction Problems. Advances in Artificial Intelligence Theory 16, 35–44 (2005)
24. Terashima-Marín, H., Farías-Zárate, C.J., Ross, P., Valenzuela-Rendón, M.: A GA-Based Method to Produce Generalized Hyper-heuristics for the 2D-Regular Cutting Stock Problem. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation, Seattle, Washington, USA, pp. 591–598 (2006)
25. Wilson, R.A., Keil, F.C.: The MIT Encyclopedia of the Cognitive Science. MIT Press, Cambridge (1999)

Optimal Design of Multiproduct Batch Plants under Imprecise Demand

Alberto A. Aguilar-Lasserre, Guillermo Cortes Robles, Blanca E. González Sánchez, Magno A. González Huerta, and Oscar Baez Senties

Division of Research and Postgraduate Studies
Instituto Tecnológico de Orizaba.
Av. Instituto Tecnológico 852, Col. Emiliano Zapata. 09340 Orizaba, Veracruz, México
aaguilar@itorizaba.edu.mx, {gc_robles,blancags67}@hotmail.com,
{magnogh,obaez70}@yahoo.com.mx

Abstract. In this study, we have introduced a fuzzy decision-making approach to design a multi-objective optimal design problem of a multiproduct batch chemical plant. The design of such plants necessary involves how equipment may be utilized, which means that plant scheduling and production must form an integral part of the design problem. This work proposes an alternative treatment of the imprecision (demands) by using fuzzy concepts. In this study, we introduce a new approach to the design problem based on a multi-objective genetic algorithm, taking into account simultaneously four criteria, i.e. maximization of the revenue and minimization of the investment cost, the operation cost and the total production time. The genetic algorithm approach was chosen since it is particularly well-suited to take into account the arithmetic of fuzzy numbers.

Keywords: Multi-objective Optimization, Genetic Algorithm, Fuzzy Arithmetic, Batch Design, Fuzzy decision-making.

1 Introduction

There has been an increased interest in the design of batch plant in recent years due to the growth of specialty chemicals, food products and pharmaceutical industries. In the conventional optimal design of a multiproduct batch chemical plant, a designer specifies the production requirements for each product and total production time. The number, required volume and size of parallel equipment units in each stage are to be determined in order to minimize the investment cost [7]. However, it is often the case that no precise product demand predictions are available at the design stage [16].

The market demand for such products is usually changeable, and at the stage of conceptual design of a batch plant, it is almost impossible to obtain the precise information on the future product demand over the lifetime of the plant. Nevertheless, decisions must be taken on the plant capacity. This capacity should be able to balance the product demand satisfaction and extra plant capacity in order to reduce the loss on the excessive investment cost or than on market share due to the varying product demands [3].

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 418–428, 2008.

The key point in the optimal design of batch plants under imprecision concerns modeling of demand variations. The most common approaches treated in the dedicated literature represent the demand uncertainty with a probabilistic frame by means of Gaussian distributions. Yet, this assumption does not seem to be a reliable representation of the reality, since in practice the parameters are interdependent, leading to very hard computations of conditional probabilities, and do not follow symmetric distribution rules. In this work, fuzzy concepts and arithmetic constitute an alternative to describe the imprecise nature on product demands.

In this study, we will introduce a fuzzy decision-making approach to solve the fuzzy goal optimization problem. For this purpose, we extended a multiobjective genetic algorithm, developed in previous works [1]. For instance, the optimal design of a multiproduct batch chemical plant is not only to minimize the investment, but also to minimize the operation cost, to minimize the total production time and to maximize the revenue, simultaneously. The paper is organized as follows. Section 2 presents a brief overview design approaches under uncertainty. Section 3 is devoted to process description, problem formulation and presents an overview of fuzzy set theory involved in the fuzzy framework within a multiobjective genetic algorithm. The presentation is then illustrated by some typical results in Section 4. Finally, the conclusions on this work are draw.

2 Design Approaches under Uncertainty

The literature offers three basic approaches to the problem of design under uncertainty [17]. They can be classified within the wait-and-see approach, the probabilistic model and the two-stage formulation.

In the wait-and-see formulation, a separate optimal design is found for each realization of the set of uncertain parameters. The cost of the plant is then calculated as the expected value of the separate designs. The practical difficulties with this model are that it is generally difficult to identify the design which yields a value of the plant cost which is equal to the expected value, and even if such a design can be identified there is usually no direct way establishing to what extent this design will accommodate other values of the uncertain parameters.

In the probabilistic model, also often called the chance constrained model, a probability of constraint satisfaction must be specified by the designer. As show by Charnes and Cooper [4], if normal distributions are assumed, if the uncertain parameters linearly occur in the constraints and if the constraints can be considered independently, then the probabilistic constraints can be reduced to a deterministic form and thus the model converts to an ordinary deterministic optimization problem.

In the two-stage formulation, also called the “here-and-now” model, the design variables are selected “here and now” so as to accommodate any future uncertain parameter realizations or perhaps those which fall within some specified confidence limits. The equipment sizes are determined at the first step or design stage and the effect of the uncertain parameters on systems performance is established in the second or operating stage. The second stage is, of course, the most important part of model since this is the stage at which the flexibility of the design is checked, possibly by including considerations of variations of the operating variables to accommodate the

uncertain parameter realizations. Since this part of the model is also the most computationally demanding, researchers have sought to reduce the computational burden by proposing various alternatives fully solving the second-stage problem [20]. Two-stage stochastic programming approaches have also been applied for design under uncertainty ([3], [6], [8], [14])

3 A Fuzzy Decision-Making Approach for Multiproduct Batch Plant Design

3.1 Problem Statement

The common approach used by previous research is to describe an optimal design problem of a multiproduct batch chemical plant as a single-objective optimization problem. In the single-objective optimal design problem, the number and size of parallel equipment units in each stage as well as the location and size of intermediate storage are to be determined in order to minimize the investment. Such an approach formulates the optimal design problem as a single-objective mixed-integer nonlinear programming (MINLP) problem ([5], [9], [11], [12], [13], [18], [19]). Indeed, the minimizing investment is not a unique objective for the optimal design problem. In real application, designers not only consider to minimize the investment, but also to minimize the operation cost and total production time, and to maximize the revenue, simultaneously.

3.2 Assumptions

The model used in this study is derived and adapted from that proposed by Modi and Karimi [11]. Actually, this model has its early roots in a work of Grossmann and Sargent [5], who developed a simple formulation for multiproduct batch plant design. This same formulation was adopted and extended by several authors such as Modi and Karimi, who, in addition to the typical batch items, considered semi-continuous stages and storage tanks (with fixed location in the process) to the initial formalism. In a majority of these mono-objective formulations, the typical criterion consists of the minimization of the investment cost corresponding to all items. Furthermore, the solution is subject to a major constraint, forcing the synthesis time of all products to respect a time horizon H. The decision variables, which define the plant configuration, are the (continuous) size and (discrete) number of the items for each processing stage.

The model used in this paper, although based in the above-mentioned formulation, was modified according to the proposal of Aguilar-Lasserre et al. [2].

The model is based on the following assumptions:

(i) The devices used in a same production line cannot be used twice by one same batch.
(ii) The production is achieved through a series of single product campaigns.
(iii) The units of the same batch or semi-continuous stage have the same type and size.
(iv) All intermediate tank sizes are finite.

(v) If a storage tank exists between two stages, the operation mode is "Finite Intermediate Storage". If not, the "Zero-Wait" policy is adopted.
(vi) There is no limitation for utility.
(vii) The cleaning time of the batch items is included into the processing time.
(viii) The item sizes are continuous bounded variables.

3.3 Model Formulation

The model considers the synthesis of *I* products treated in *J* batch stages and *K* semi-continuous stages. Each batch stage consists of *mj* out-of-phase parallel items with same size *Vj*. Each semi-continuous stage consists of *nk* out-of-phase parallel items with same processing rate *Rk* (i.e. treatment capacity, measured in volume unit per time unit). The item sizes (continuous variables) and equipment numbers per stage (discrete variables) are bounded. The *S-1* storage tanks, with size *Vs**, divide the whole process into *S* sub-processes.

3.3.1 Objective Function

Multi-objective optimization provides a framework for understanding the relationships between the various objective functions and allows an engineer to make decisions on how to trade-off amongst the objectives to achieve performance considered "the best". If the objective functions are incommensurate, or competing, then minimization of one objective function requires a compromise in another objective function. Such an optimal design problem becomes a MOOP as following (maximization of the revenue and minimization of the investment cost, the operation cost and the total production time):

$$Max\ (V_p) = \sum_{i=1}^{N} C_{Pi} Q_i \tag{1}$$

$$Min\ (Cost) = \sum_{j=1}^{J} (m_j a_j V_j^{\alpha_j}) + \sum_{k=1}^{K} (n_k b_k R_k^{\beta_k}) + \sum_{s=1}^{S} (c_s V_s^{\gamma_s}) \tag{2}$$

$$Min\ (D_p) = \sum_{i=1}^{N} \sum_{j=1}^{M} C_{E_j} \frac{Q_i}{B_{is}} + C_{O_i} Q_i \tag{3}$$

$$Min\ (H) = \sum_{i=1}^{I} \frac{Q_i}{B_{is}} T_{is}^{L} \tag{4}$$

3.3.2 Constraint Formulation

The problem statement involves four forms of different constraints as reported in literature [11]:

(i) Dimension constraints: every units has to restrict to its allowable range.

$$V_{min} \le V_j \le V_{max} \qquad \forall j \in \{1, \ldots, J\} \tag{5}$$

$$n_{min} \le N_j \le n_{max} \qquad \forall j \in \{1, \ldots, J\} \tag{6}$$

(ii) Time constraint: the summation of available production time for all products is inferior to the total production time.

$$\sum_{i=1}^{I} \frac{Q_i}{B_{is}} T_{is}^{L} \le H \tag{7}$$

(iii) Limiting cycle time for product *i*.

$$\frac{T_{is}{}^{L}}{n_{ij}} \leq T_{ij} \qquad \forall\, i \in \{1,\ldots,I\} \quad \forall\, j \in \{1,\ldots,J\} \tag{8}$$

(iv) Volume constraints: the volume V_j has to be able to process all the products *i*.

$$S_{ij} B_i \leq V_j \qquad \forall\, i \in \{1,\ldots,I\}, \forall\, s \in \{1,\ldots,S\} \tag{9}$$

3.4 Study Antecedent: Fuzzy Logics

The emergence of electronic commerce and business to business applications has, in a recent period, considerably changed the dynamics of the supplier-customer relationship. Indeed, customers can change more rapidly their orders to the suppliers and many enterprises have to organize their production even if the demand is not completely known at short term. On the other hand, the increasing need for integration and optimization in supply chains leads to a greater sensitivity to perturbations resulting of this uncertainty. These two elements clearly show the interest of taking into account as soon as possible the uncertainty on the demand and to propagate it along the production management mechanisms.

In the context of engineering design, an imprecise variable is a variable that may potentially assume any value within a possible range because the designer does not know *a priori* the final value that will emerge from the design process. The fuzzy set theory was introduced by Zadeh [21] to deal with problems in which a source of vagueness is involved. It is well-recognized that fuzzy set theory offers a relevant framework to model imprecision.

Huang and Wang [7] have introduced a fuzzy decision-making approach to design a multi-objective optimal design problem of a multiproduct batch chemical plant. A monotonic increasing or decreasing membership function is used to define the degree of satisfaction for each objective function.

The proposed approach involves arithmetic operations on fuzzy numbers and quantifies the imprecision of the demand by means of fuzzy sets (trapezoidal). In this case, the flat line over the interval (q_2,q_3) represents the precise demands with an interval of confidence at level α=1, while the intervals (q_1,q_2) and (q_3,q_4) represent the "more or less possible values" of the demand. The result of the *total production time* $\tilde{H}_i$, the revenue and the operation cost are treated and analyzed through fuzzy numbers. The demand and the production time are fuzzy quantities as shown in figure 1.

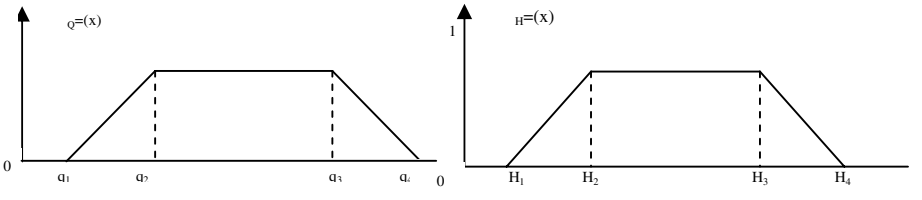


Fig. 1. Two trapezoidal fuzzy numbers, $\tilde{Q} = (q_1,q_2,q_3,q_4)$ and H=(H_1,H_2,H_3,H_4)

The criterion minimizes the delays and advances of the production time necessary for the synthesis of all the products. We must compare the time horizon $\tilde{H}$ represented by a fuzzy expression (rectangle) and the production time $\tilde{H}_i$ (trapezoidal). For the comparison of fuzzy numbers, Liou and Wang's method [10] was adopted.

The production time necessary to satisfy each product demand must be less than a given time horizon, but due to the nature of the fuzzy numbers, three different cases for determination of the criterion may occur. The different cases are reported in figure 2.

The temporal criterion selected is called "common surface", representing the intersection between the sum of the production time (trapezoid) and the horizon of time to respect (rectangle). The calculation of the criterion depends on each case: for example, case1 illustrate the solutions which arrive just in time.

3.5 Overview of Multiobjetive Genetic Algorithm Approach

General principles on evolutionary computation will not be recalled here, since the aim is only to insist on the particular details of the used Genetic Algorithm (GA). Actually, this one is a standard multiobjective algorithm, similar to the specific one developed in [1]. The proposed GA procedure implies the following steps:

1) ***Encoding of solution.*** The encoding of the solutions was carried out dividing the chromosome, i.e. a complete set of coded variables, into two parts. The first one deals with the items volumes, which are continuous in the initial formulation. Nevertheless, they were discretized here with a 50 unit range, while their upper and lower bounds

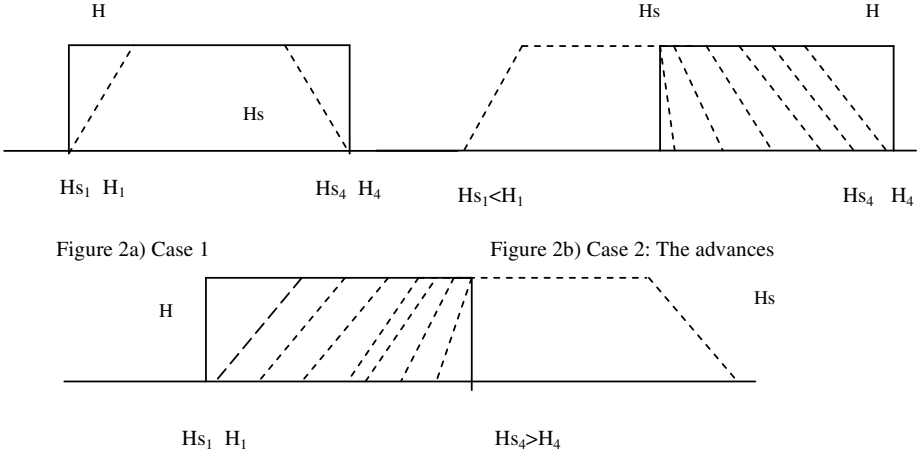


Fig. 2. Three cases for the minimization of a criterion that penalizes the delays and advances of the time horizon necessary for the synthesis of all the products

were preserved. The second part of the chromosome handles the number of equipment items per stage: the value of these discrete variables is coded directly in the chromosome.

2) ***Initial population creation.*** The procedure of creating the initial population corresponds to random sampling of each decision variable within its specific range of variation. This strategy guarantees a population various enough to explore large zones of the search space.

3) ***Fitness Evaluation.*** The optimization criterion considered for fitness evaluation involves the *revenue* and three other performance criteria, i.e. *the production delay/advance, the investment and the operation cost.*

4) ***Selection Procedure.*** The multi-objective aspects are taken into account in the selection procedure. The better individuals, that are the surviving individuals, are chosen with the Goldberg's roulette. The same number of surviving individuals is chosen for each criterion (four roulettes). Each individual is represented by a slice of the roulette wheel, proportional to its fitness value. Since the criteria are represented by fuzzy numbers, they were defuzzified (the defuzzified value was calculated as the centre of gravity) in the roulette wheel.

5) ***Crossover and mutation.*** Concerning genetic operators, the crossover obeys to a classical one cut-point method, while the mutation technique is in agreement with the encoding procedure: inversion of a bit value on the continuous zone (0 becomes 1 and conversely), and decrease of one unit for the discrete loci (if possible, of course).

6) ***Registration of all non-dominated individuals in Pareto set.*** Pareto's sort procedure is carried out at the end of the algorithm over all the evaluated solutions; at the end of the procedure, the whole set of the non dominated Pareto's optimal solutions, are obtained.

3.6 Treatment of an Illustrative Example

We consider an example to illustrate the approach fuzzy-AG based on arithmetic operations on fuzzy numbers and quantifying the imprecision of the demand. The example was initially presented by Ponsich and al. [15]: the plant, divided into two sub-processes, consists of six batch stages to manufacture three products.

The GA parameters are the following ones: Population size 400 individuals, number of generations 500 iterations, crossover probability 40%, mutation probability 30% and the stop criterion considered in this study concerns a maximum number of generations to reach.

For the considered example, table 1 shows the values for processing times, size factor for the units, cost data, and the production requirement for each product quantifying the imprecision of the demand by means of fuzzy numbers representing the "more or less possible values".

For the construction of the trapezoid which represents the request for each product, the original values of the demand were used as a reference. To determine the support and the core, one calculated a percentage of opening taking as reference the demand of the original data is computed.

Table 1. Data used in example

		Processing time $t_{i,j}$ *(h)*						*Size factors (1/kg)*					
		B1	*B2*	*B3*	*B4*	*B5*	*B6*	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*
Minimum size =250 l	**A**	1.15	3.98	9.86	5.28	1.2	3.57	8.28	6.92	9.7	2.95	6.57	10.6
Maximum size = 10 000 l	**B**	5.95	7.52	7.01	7	1.08	5.78	5.58	8.03	8.09	3.27	6.17	6.57
	C	3.96	5.07	6.01	5.13	0.66	4.37	2.34	9.19	10.3	5.7	5.98	3.14
	t_j	0.4	0.29	0.33	0.3	0.2	0.35						

Unit price for product i ($/Kg)			*Coefficients* $c_{i,j}$						Q_1=(428260, 432630, 441370, 445740) Q_2=(317520, 320760, 327240, 330480) Q_3=(252840, 255420, 260580, 263160) H = (5800, 5800, 6200, 6200) The horizon *H* (rectangle) (h)
	C_P	C_O	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	
A	0.70	0.08	0.2	0.36	0.24	0.4	0.5	0.4	Cost of mixer=250V^{0.6}$
B	0.74	0.1	0.15	0.5	0.35	0.7	0.42	0.38	Cost of reactor=250V^{0.6}$
C	0.80	0.07	0.34	0.64	0.5	0.85	0.3	0.22	Cost of centrifuge=250V^{0.6}$ (Volume V in liter)

Operating cost factors						
	B1	*B2*	*B3*	*B4*	*B5*	*B6*
C_E	20	30	15	35	37	18

4 Typical Results

The result obtained of the multi-objective optimization problem involves 44 non-dominated solutions: among them, 42 represent the first case, where the production time H_i (trapezoidal) is within of time horizon H (rectangle) The solutions 43 and 44 represent respectively cases 2 and 3 (the criterion that penalizes the delays and advances of the time horizon necessary for the synthesis of all the products). This study leads to three different scenarios as a fuzzy decision-making approach to preliminary design.

Table 2 shows the solution that minimizes the investment cost for case 1. Tables 3 and 4 present the results that minimize the penalization of delays and advances (cases 2 and 3), respectively. These tables display typical results obtained after ten runs to guaranty the stochastic nature of the algorithm. Since a multi-objective analysis is considered, a statistical analysis of the results is difficult to carry out and was not performed as it was the case in our previous studies in mono-objective optimization.

From these results, it can be concluded that:

(i) The fuzzy optimal design of batch plant is the solution of case 2 (the advance) because its operation cost is lower than the solution of case 1. The production time of the case of advance $\tilde{H}_i$ = [5788.7, 5847.8, 5965.9, 6025] is lower than case 1 $\tilde{H}_i$ = [5800.1, 5859.3, 5977.7, 6036.9].

(ii) The fuzzy design of batch plant for case 1 is a sure solution because the production time $\tilde{H}_i$ = [5800.1, 5859.3, 5977.7, 6036.9] is within the imposed time horizon $\tilde{H}$ = [5800, 5800, 6200, 6200].

(iii) The fuzzy design of batch plant for the case 3 is a more hazardous solution, because the production time $\tilde{H}_i = [5958.5,\ 6019.3,\ 6140.9,\ 6201.7]$ is greater than the time horizon $\tilde{H} = [5800, 5800, 6200, 6200]$.

Table 2. Fuzzy optimal design of batch plant for case 1

Product	B_{is} kg	T_{Li} h	*Optimal objective function value*	*Storage Tanks*
A	589.6	3.9	I = 721078.1 [$]	V_s = 2000.9 [l]
B	951.2	5.8	$\tilde{V}_p$ = [737018.8 744539.4 759580.6 767101.2] [$]	
C	873.7	5.0	$\tilde{D}_p$ = [270064.4 272820.2 278331.7 281087.5] [$]	
			$\sum \tilde{H}_i$ = [5800.1 5859.3 5977.7 6036.9] [h]	

Table 3. Fuzzy optimal design of batch plant for the criterion that penalizes the advances

Product	B_{is} kg	T_{Li} h	*Optimal objective function value*	*Storage Tanks*
A	832.3	4.4	I = 693084 [$]	V_s = 2121.9 [l]
B	1008.7	6.6	$\tilde{V}_p$ = [737018.8 744539.4 759580.6 767101.2] [$]	
C	771.9	5.8	$\tilde{D}_p$ = [251027.7 253589.2 258712.9 261273.0] [$]	
			$\sum \tilde{H}_i$ = [5788.7 5847.82 5965.9 6025] [h]	

Table 4. Fuzzy optimal design of batch plant for the criterion that penalizes the delays

Product	B_{is} kg	T_{Li} h	*Optimal objective function value*	*Storage Tanks*
A	943.5	6.2	I = 698906.2 [$]	V_s = 2150.6 [l]
B	1226.6	6.3	$\tilde{V}_p$ = [737018.8 744539.4 759580.6 767101.2] [$]	
C	964.9	5.7	$\tilde{D}_p$ = [232624.1 234997.8 239745.27 242118.9] [$]	
			$\sum \tilde{H}_i$ = [5958.5 6019.3 6140.9 6201.7] [h]	

5 Conclusions

For the most common form of design of a multiproduct batch chemical plant, the designers specify the production requirement of each product and the total production time. However, no precise product demand predictions are generally available. For this reason, an alternative treatment of the imprecision by using fuzzy concepts is introduced in this paper.

In real application, designers not only require to minimize investment cost, but also to minimize the operation cost and the delays and advances of the time horizon necessary for the synthesis of all the products, and to maximize the revenue, simultaneously.

In this study, we have introduced a fuzzy-AGs approach to solve the problem of multi-objective optimal design of a multiproduct batch chemical plant. The results obtained on the treated example have shown that three different scenarios were obtained as a fuzzy decision-making approach. The analysis tended to be helpful for decision making.

Nomenclature
C_{Ej}= Operation cost in stage j(\$)
C_{Oi}= Operation cost of product i to be produced (\$/kg)
C_{Pi}= Price of product i (\$/kg)
S_{ij}= Size factor of product i in stage j (1/kg)
B_{is}= Size of the batch of product i
Q_i= Production requirement of product i (kg)
T_{Li}= Cycle time for product i (h)
α_j= Cost coefficient for unit j
β_j= Cost exponent for unit j
τ_{ij}= Processing time of product i in stage j (h)

References

1. Aguilar Lasserre, A.A., et al.: Modélisation des imprécisions de la demande en conception optimale multicritère d'ateliers discontinus. In: Proceedings of the SFGP (Société Français de Génie de Procédés), Toulouse, France (2005)
2. Aguilar-Lasserre, A.A., et al.: Enhanced Genetic Algorithm-based Fuzzy Multiobjective Strategy to Multiproduct Batch Plant Design. In: Proceedings of the International Fuzzy Systems Association world congress, Cancun, México, June 18-21 (2007)
3. Cao, D., Yuan, X.: Optimal design of batch plants with uncertain demands considering switch over of operating modes of parallel units. Ind. Eng. Chem., 4616–4625 (2002)
4. Charnes, A., Cooper, W.: Chance constrained programming. Mgmt Sci., 73–78 (1959)
5. Grossmann, I.E., Sargent, R.W.H.: Optimum design of multipurpose chemical plants. Industrial Engineering and Chemical Process Design and Development 18(2), 343–348 (1979)
6. Harding, S.T., Floudas, C.A.: Global optimization in multiproduct and multipurposebatch design under uncertainty. Industrial Engineering and Chemistry Research 36, 1644 (1997)
7. Huang, H., Wang, W.F.: Fuzzy decision-making design of chemical plant using mixed-integer hybrid differential evolution. Computers & Chemical Engineering 26(12, 15), 1649–1660 (2002)
8. Ierapetritou, M.G., Pistikopolous, E.N.: Batch plant design and operations under demand uncertainty. Industrial Engineering and Chemistry Research 35, 772 (1996)
9. Knopf, F.C., Okos, M.R., Reklaitis, G.V.: Optimal design of batch/semicontinuous processes. Industrial Engineering and Chemical Process Design and Development 21(1), 79–86 (1982)
10. Liou, T.S., Wang, M.J.: Ranking fuzzy numbers with integral value. Fuzzy Sets System 50, 247 (1992)
11. Modi, A.K., Karimi, I.A.: Design of multiproduct batch processes with finite intermediate storage. Computer and Chemical Engineering 13, 127–138 (1989)
12. Montagna, J.M., Vecchietti, A.R., Iribarren, O.A., Pinto, J.M., Asenjo, J.A.: Optimal design of protein production plants with time and size factor process models. Biotechnology Programming 16, 228–237 (2000)
13. Patel, A.N., Mah, R.S.H., Karimi, I.A.: Preliminary design of multiproduct non-continuous plants using simulated annealing. Computers and Chemical Engineering 15(7), 451–469 (1991)
14. Petkov, S.V., Maranas, C.D.: Design of single-product campaign batch plants under demand uncertainty. AIChE Journal 44, 896 (1998)

15. Ponsich, A., et al.: About the Relevance of Mathematical Programming and Stochastic Optimisation Methods: Application to Optimal Batch Plant Design Problems. Escape 15 (2004)
16. Shah, N., Pantelides, C.C.: Design of multipurpose batch plants with uncertain production requirements. Ind. Eng. Chem. Res., 1325–1337 (1992)
17. Vajda, S.: Probabilistic Programming. Academic Press, London (1972)
18. Wang, C., Quan, H., Xu, X.: Optimal design of multiproduct batch chemical process using genetic algorithms. Industrial and Engineering Chemistry Research 35(10), 3560–3566 (1996)
19. Wang, C., Quan, H., Xu, X.: Optimal design of multiproduct batch chemical process using tabu search. Computers and Chemical Engineering 23(3), 427–437 (1999)
20. Wellons, H.S., Reklaitais, G.V.: The design of multiproduct batch plants under uncertainty with stage expansion. Computers chem. Engng. 13(1/2), 115–126 (1989)
21. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning. Information Sciences 8(3), 199–249 (1975)

A Set of Test Cases for Performance Measures in Multiobjective Optimization

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. Comparing the performance of different evolutive multiobjective algorithms is an open problem. With time, many performance measures have been proposed. Unfortunately, the evaluations of many of these performance measures disagree with the common sense of when a multiobjective algorithm is performing better than another. In this work we present a benchmark that is helpful to check if a performance measure actually has a good behavior. Some of the most popular performance measures in literature are tested. The results are valuable for a better understanding of what performance measures are better.

1 Introduction

Evolutionary Multiobjective Optimization (*EMO*) consists of minimizing a vector of objective functions $F(x) = \langle f_1(x), f_2(x), \ldots, f_n(x)\rangle$ subject to constraints using Evolutionary Algorithms (EA). The objective functions and constraints depend on a vector of variables $x \in \mathbf{R}^m$. We call X the set of all vectors x that do not violate the constraints and Z the projection of X on objective functions space. Without loss of generality, we consider hereafter that we are minimizing the objective functions.

EMO uses the Pareto Optimality Criteria (*POC*). POC is defined through the relation between two vectors $z, w \in \mathbf{R}^n$ known as Pareto dominance. We have that z dominates w ($z \succeq w$) if $\forall i \in \{1, 2, \ldots, n\}, z_i \leq w_i \ \wedge \ \exists j \in \{1, 2, \ldots, n\} \mid z_j < w_j$, where v_k stands for the k–th component of vector v. The goal is to find a set of vectors known as the Pareto Set (*PS*) defined as $PS = \{x \in X \mid \forall y \in X, F(y) \not\succeq F(x)\}$. According to POC, all elements of PS are optimal, because they represent the different tradeoffs between the objective functions where it is not possible to improve one objective without degrading another. The projection of PS in objective functions space is called the Pareto Front (*PF*), and is usually described as a surface that represents the best tradeoff possible between the objective functions.

In recent years, many multiobjective evolutionary algorithms (MOEAs) based on POC have been developed. For a review of these algorithms, consult [4] and [3]. Instead of generating a single solution, these algorithms generate a set A of vector solutions $x \in X$ that approximate the PS. These approximation sets

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 429–439, 2008.

have the characteristic that $\forall x, y \in A, F(x) \not\succeq F(y) \ \wedge \ F(y) \not\succeq F(x)$. These sets are usually known as non–dominated sets (NS).

One of the most important difficulties about using the POC is how to compare the performance of different algorithms. Usually, this is done by comparing the NSs that the algorithms generate. So, many performance measures (we also use the term quality indicator, QI or metrics) that evaluate the quality of NSs have been proposed. A consequence of this is that a new problem has arisen, what performance measure must we use to evaluate the NSs? When using a QI we want to create an order (or a partial order) between the non–dominated sets we are evaluating. For example, for NSs A, B, C, D, E we want that a QI allows us to establish an order like $C > B > A > D > E$ where $A > D$ means that A is better than D. Unfortunately, it is common that different QIs give different orderings, so the ordering depends on the performance measure we are using. Further, some QIs give orderings that completely contradict our expectations. For example, if we know that set $A > B$ based on our knowledge of the multiobjective problem, it is possible for some performance measures to evaluate B as better than A.

The conclusion is that some quality indicators are better than others. In order to find what QI are better, some theoretical studies have been done [7] [10], but still there is no definitive performance measure to evaluate evolutionary multiobjective algorithms. So, we can expect that more QIs will be designed in the future. Considering all this, it is surprising that no test cases have been proposed to evaluate the metrics, as there are many benchmarks to evaluate multiobjective algorithms. In this work we try to fill this gap and present a set of test cases for QIs.

The organization of the rest of the paper is as follows. In Section 2 we introduce some basic concepts about multiobjective optimization and performance measures. In Section 3 we review some of the performance measures available in the literature. In Section 4 we present our test cases with detailed descriptions of their characteristics. The results and analysis of the evaluation of some performance measures is presented in Section 5. Finally, in Section 6 we state our conclusions based on the result of the experiments.

2 Basic Concepts

A MOEA generates a non–dominated set that approximates the Pareto Front. In order to create a performance measure, we need to define what we want from a non–dominated set. Usually we focus in these two properties:

Convergence: it refers to how close is a NS to the PF. It comes directly from the definition of the Pareto's optimality.

Diversity: we are interested in the different tradeoffs of the objective functions in a multiobjective problem. As the output of a multiobjective algorithm is a finite set of solutions, it is desirable to maximize the information these solutions provide and avoid the situation where there are zones of the Pareto Front with too many solutions and zones with too few.

Most of the research in performance measures for NSs is focused on convergence. A classical work is that of Hansen and Jaszkiewicz [6], where they define the three following relationships between two NSs A and B.

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 $a \succeq b$ or $a = b$ 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 $a \succeq b$ or $a = b$ and there exists at least a pair of points $r \in A$ and $s \in B$ such that $r \succeq s$.

Complete outperformance: A completly outperforms B ($A\ O_C\ B$), if for every point $b \in B$ there exists at least one point $a \in A$ so that $a \succeq b$.

These outperformance relations are used to establish a minimum of what we expect from a comparison method. It is easy to understand that $A\ O_C\ B$ implies that A is a better than B, because for every vector in B there is a better one in A. So, if we have a comparison method R, and it evaluates B as better than A, then R is not reliable. Similar conclusions can be obtained form O_S and O_W. With all this in mind, Hansen and Jaszkiewicz [6] also define the property of *compatibility with an outperformance relation* O, where O can be O_W, O_S or O_C, as follows:

Compatibility. A comparison method R is (weakly) compatible with O if $A\ O\ B$ implies that R will evaluate A as (not worse) better than B. In other words, $A\ O\ B \Longrightarrow C(A > B)$.

The compatibility with the outperformance relations is desirable because it makes a comparison method more robust to misleading cases, and with a behavior according to intuition.

Zitzler et al [10] made a study based on convergence and defined the property of "compatible and completeness" (CC), that essentially transforms the implication in the definition of compatibility, into a double implication. Compatibility and completeness established a stricter criteria of what properties are desirable for a performance measure. According to Zitzler et al [10] it is important because a compatible and complete comparison method is able to decide whether a NS is better than another.

There are many QIs in literature, some of them have the properties mentioned above and some others do not [10] [7]. There are two main classifications for QIs. The first classification is based on what characteristic of a non–dominated sets is evaluated. For example, there are metrics that only evaluate convergence of the non–dominated sets based in the concept of outperformance relations. We call these metrics "convergence only". Also, there are metrics that only evaluate the diversity of a non–dominated set. We call these metrics "diversity only". Finally, there are metrics that evaluates the general quality of a non–dominated set considering all its properties at the same time. We call these metrics "all around".

The second classification is based on the number of NSs the QI takes as argument. For example, unary metrics takes only one non–dominated set and

generates one (or more) real values. These real values are an evaluation of the quality of the NS. When comparing several NS, we only need to compare these real values to establish the order of the sets. Binary metrics, takes two non–dominated sets as argument and use all the vectors in both sets to decide which one is better than the other. As we are using all information in both NS to make the comparison we can expect a more accurate result. It was demonstrated [10] that only binary quality indicators can have the property of compatibility and completeness.

In the following section we describe some of the best know performance measures in literature.

3 A Review of the Performance Measures in Literature

Several performance measures have been proposed in the past years in order to evaluate the quality of NSs. Different measures have different characteristics and some authors (Knowles and Corne [7], Zitzler et al [10]) have analyzed their effectiveness. Now we describe some of the them.

S-metric (S). Proposed in [9], the S–metric consists on the space enclosed by a non–dominated set A and a fixed reference point r^*. For a minimization problem, the bigger the space the better is A. It is a unary and all around performance measure. The election of the reference point is vital for the good behavior of the S–metric. It has a bias toward the central zone and convex zones of the Pareto Front [2]. The computational complexity is $O(|A|^{(n/2)})$ for more than 3 objective functions [1] and $O(|A|log|A|)$ for 2 or 3 objective functions [5]. The S–metric is compatible with all the outperformance relations, and has an intuitive meaning.

Schott's Spacing Metric (SS). Designed by Schott [8] to measure the spread of the points in space, SS is calculated with the formula $SS(A) = \sqrt{\sum_{i=1}^{|A|}(d^* - d_i)^2/(|A|-1)}$, where A is the non–dominated set we are evaluating, d_i is the distance between the element i of A and the element of A nearer to it, d^* is the mean of d_i. It is a unary and diversity only performance measure. It is not compatible with the outperformance relations. It can be useful when combined with other measures. It is easy to calculate with a low computational complexity ($O(|A|^2)$).

Epsilon Indicator (I_ϵ). Proposed in [10], $I_\epsilon(A, P^*)$ is the minimum factor for which we need to multiply all the elements of a reference set P^* in order to have all its elements dominated or equal to the elements in a non–dominated set A. Smaller values of $I_\epsilon(A, P^*)$ means that the set A is more similar to the reference set and that it is a better NS (because the reference set is suppose to be a better NS than any set in the comparison). It is compatible with all the outperformance relations. It is easy to calculate and has a low computational complexity ($|A| \cdot |P^*| \cdot n$). It is a unary and all around QI.

Binary Epsilon Indicator (B_ϵ). Proposed by Zitzler et al [10], the Binary Epsilon Indicator is a binary performance measure based on the Epsilon Indicator.

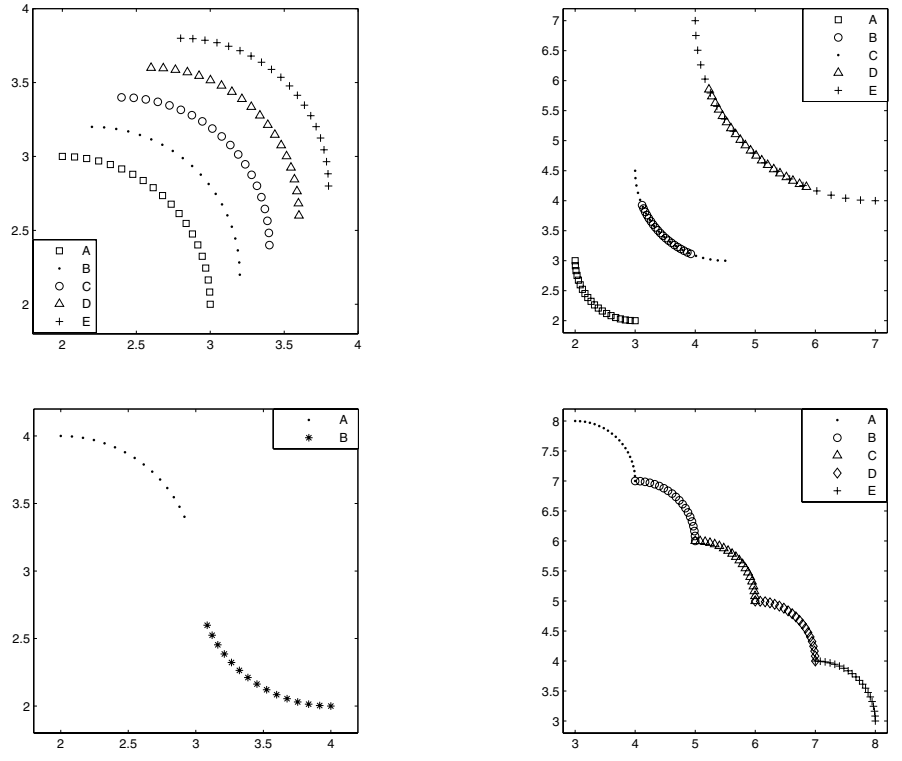


Fig. 1. Test cases 1, 2, 6 and 7

For two NSs A and B, it is defined as $B_\epsilon(A, B) = (I_\epsilon(A, B) = 1) \wedge (I_\epsilon(B, A) > 1)$, returning a true value if A outperforms B, and a false value otherwise. It is a convergence only quality indicator and it is compatible with all the outperformance relations. It is easy to calculate and essentially has the same computational complexity than I_ϵ. It is a convergence only QI. To the knowledge of the authors, this is the only compatible and complete quality indicator in literature.

4 The Benchmark

In this section we present several test cases that can be used to test performance measures. We constructed these test cases in such a way that it is evident what non–dominated set is better than other. The challenge for the performance measure is to find the correct order between the different non–dominated sets. In each test case there are five NSs (A, B, C, D and E), except for case 6 where we have two NSs (A and B). We created a 2 dimensional version and a 3 dimensional version for each test case.

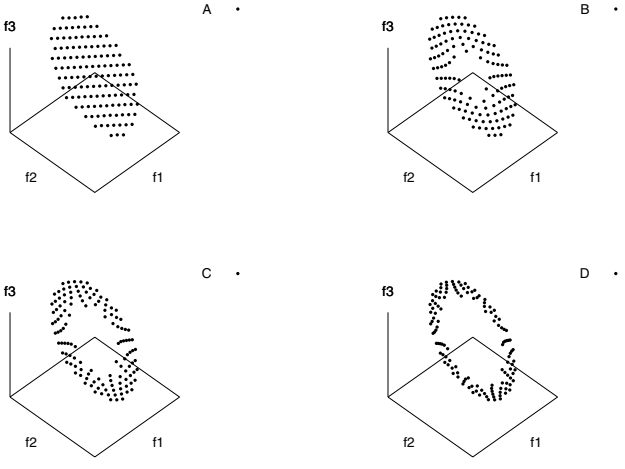


Fig. 2. Test case 3

The nomenclature we use to represent the different relationship between different non–dominated sets is the following. Suppose we have five NSs A, B, C, D and E. $A > B$ means that A is better than B, $A = B$ means that A and B are equally good, A ? B means that A and B are not comparable under the outperformance relations. $A > B, C, D$ means that A is better than any of B, C and D; and $A, B, C > E$ means that E is worse than any of A, B, and C.

For example, $E > D > C > B > A$ means that E is better than any other sets; D is better than A, B and C; C is better than A and B; and B is better than A.

$E > D$? $C > B$? A means that E is better than any other set; D and C are better than A and B but no comparable between them; and B and A are not comparable.

$E = D = C = B > A$ means that D, B, C and D are equally good and that A is worse than any of the other NSs.

The ordering we expect from a quality indicator depends on what characteristic of the non–dominated sets is being evaluated. The expected ordering from a convergence only QI may be different from the expected ordering from an all around QI. In the description of the test cases we specify the expected ordering depending on the kind of performance measure.

Test Case 1. This case focus on convergence. The sets were constructed in such a way that $A\ O_C\ B$; $B\ O_C\ C$; $C\ O_C\ D$ and $D\ O_C\ E$. All sets have the same number of elements, dispersion and extension. So, the expected ordering for a convergence only QI and for an all around QI is $A > B > C > D > E$. For a diversity only QI the expected ordering is $A = B = C = D = E$. The 2d version is shown in Figure 1, top left.

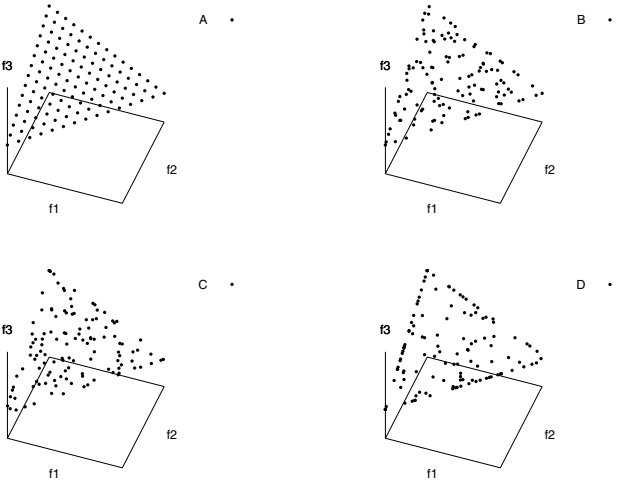


Fig. 3. Test case 4

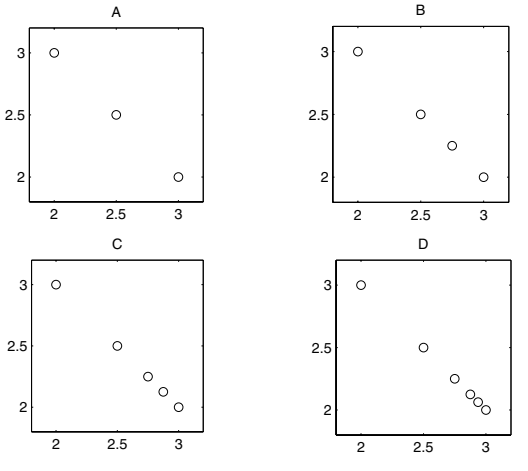


Fig. 4. Test case 5

Test Case 2. In this case we combined convergence with diversity. The sets were constructed in such a way that $A\ O_C\ B, C$; $B, C\ O_C\ D, E$; B and C do not outperform each other; D and E do not outperform each other. All sets have the same number of elements and dispersion but E has a better extension than D, D has a better extension than C, C has a better extension than A and A has a slightly better extension than B. So, the expected ordering for a convergence only QI is $A > B\ ?\ C > D\ ?\ E$; for an all around QI the ordering is $A > C > B > E > D$; and for a diversity only QI the ordering is $E > D > C > A > B$. The 2d version is shown in Figure 1, top right.

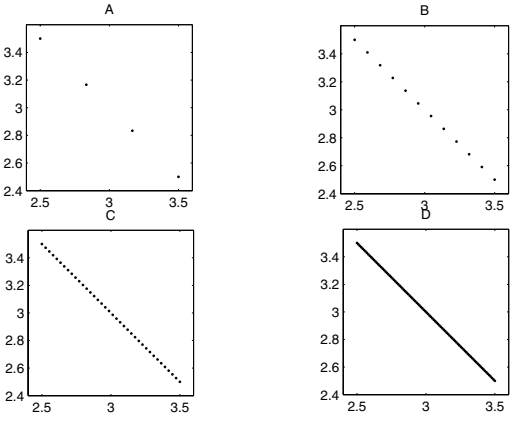


Fig. 5. Test case 8

Test Case 3. The objective for this case is to evaluate if the QIs can detect holes inside the non–dominated sets. The sets have the same number of vectors and the same convergence, but they have an empty space inside them. The size of the empty space is different depending on the set. We assume that the bigger of the hole, the worse the quality of the NS. So, the expected ordering for an all around QI and a diversity only QI is $A > B > C > D > E$; for a convergence only QI the ordering is $A\ ?\ B\ ?\ C\ ?\ D\ ?\ E$. The 3d version of sets A, B, C and D is shown in Figure 2.

Test Case 4. We evaluate diversity in this test case. All sets have the same number of points, extension and convergence. Set A was constructed in such a way that its elements are evenly distributed. The rest of the sets were constructed adding an increasing amount of uniform noise to A, but without affecting its extension and convergence. So, the expected ordering for an all around QI and a diversity only QI is $A > B > C > D > E$; for a convergence only QI the ordering is $A\ ?\ B\ ?\ C\ ?\ D\ ?\ E$. The 3d version of sets A, B, C and D is shown in Figure 3.

Test Case 5. This is an interesting case because it can be used to evaluate convergence and diversity independently. In 2d, we created set A with 3 vector evenly distributed (Figure 4); B was created adding a new non–dominated vector to A; C were created adding a new non–dominated vector to B and so on. As a result, $E\ O_W\ D\ O_W\ C\ O_W\ B\ O_W\ A$ and at the same time we have that E has a better diversity than D, because it has the same information plus one more vector than D, D has the same information plus one more vector than C, etc. The 3d version were created in a similar way. So, due to the construction of the NSs, the expected ordering for all kind of quality indicators is $E > D > C > B > A$.

Test Case 6. The objective of this test case is to detect if the metrics are affected by the convexity of the pareto front. We have only two NSs, A and B. Both sets

have the same convergence, dispersion, extension and number of vectors but they have different convexity (Figure 1, bottom left). We consider that there is no preference for the convexity of the Pareto Front. The expected ordering for an all around QIs and a diversity only QIs is $A = B$; for a convergence only QI the ordering is $A \; ? \; B$.

Test Case 7. The goal in this experiment is to evaluate if the performance measures are affected by the relative position of the non–dominated sets. All sets have the same convergence, dispersion, extension and number of vectors; but they are in different parts of the Pareto Front (Figure 1, bottom rigth). The expected ordering for an all around QI and a diversity only QI is $A = B = C = D = E$; for a convergence only QI the ordering is $A \; ? \; B \; ? \; C \; ? \; D \; ? \; E$.

Test Case 8. For this experiment(Figure 5), we generated five NSs with the same extension, convergence, the vectors are evenly distributed, but the sets have a different number of vectors. So we have that $|A| = 4, |B| = 12, |C| = 38, |D| = 98, |E| = 758$ for the 2d case. For the 3d case, $|A| = 10, |B| = 21, |C| = 35, |D| = 78, |E| = 465$. The objective is to study how the number of vectors can affect the behavior of the metric. The ordering for a convergence only performance measure is $A \; ? \; B \; ? \; C \; ? \; D \; ? \; D$; for an all around and diversity only metric, the ordering is $E > D > C > B > A$.

We use the these test cases to evaluate the performance measures described in Section 3. The reference set for I_ϵ are the non–dominated vectors from the union of the NSs in the corresponding test case. The reference point for the S–metric is the vector formed by the maximum values of all vectors of all NSs in the corresponding test case. The results are presented and discussed in the next section.

5 Analysis and Results

The results of the experiments are shown in Table 1. As we can see, the S–metric is a very good performance measure, ordering the sets correctly except in Cases 6 and 7. This confirms that the S–metric has a bias toward convex zones of the Pareto front and toward the central part (with respect to the reference point) of the Pareto front. The Epsilon Indicator is not as good as the S–metric. It has the same biases than the S–metric (it failed Cases 6 and 7). Besides, it failed Case 2 and the 3d version of Case 4.

Schott Spacing metric gave the correct order in Cases 1, 5, 6, 7, 8, but in Case 2 it ordered the set in the wrong order. It also failed Cases 3 and 4. The reason for this is that the Schott Spacing metric is the standard deviation of the distance between the vectors and their nearest neighbor. But, as it is not normalized with respect to the mean, the value of the standard deviation will be bigger for sets with more extension, even when they are evenly distributed.

The Binary Epsilon Indicator ordered all sets as expected for a convergence only QI. This shows that this QI is very reliable. It is important to remember

Table 1. Results of the Experiments

Two Dimensions				
Case	I_ϵ	S	B_ϵ	SS
1	$A > B > C > D > E$	$A > B > C > D > E$	$A > B > C > D > E$	$A = B = C = D = E$
2	$A > C > B > D > E$	$A > C > B > E > D$	$A > B \ ? \ C > D \ ? \ E$	$B > A > C > D > E$
3	$A > B > C > D > E$	$A > B > C > D > E$	$A \ ? \ B \ ? \ C \ ? \ D \ ? \ E$	$D > C > B > A > E$
4	$A > B > C > D > E$	$A > B > C > D > E$	$A \ ? \ B \ ? \ C \ ? \ D \ ? \ E$	$C > D > E > B > A$
5	$E > D > C > B > A$	$E > D > C > B > A$	$E > D > C > B > A$	$E > D > C > B > A$
6	$B > A$	$B > A$	$A \ ? \ B$	$A = B$
7	$C > B = D > A = E$	$C > B = D > A = E$	$A \ ? \ B \ ? \ C \ ? \ D \ ? \ E$	$A = B = C = D = E$
8	$E > D > C > B > A$	$E > D > C > B > A$	$A \ ? \ B \ ? \ C \ ? \ D \ ? \ E$	$E > D > C > B > A$
Three Dimensions				
Case	I_ϵ	S	B_ϵ	SS
1	$A > B > C > D > E$	$A > B > C > D > E$	$A > B > C > D > E$	$A = B = C = D = E$
2	$A > C > B > D > E$	$A > C > B > E > D$	$A > B \ ? \ C > D \ ? \ E$	$B > A > C > D > E$
3	$A > B > C > D > E$	$A > B > C > D > E$	$A \ ? \ B \ ? \ C \ ? \ D \ ? \ E$	$E > D > C > B > A$
4	$A > B > D > C > E$	$A > B > C > D > E$	$A \ ? \ B \ ? \ C \ ? \ D \ ? \ E$	$E > D > B > C > A$
5	$E > D > C > B > A$	$E > D > C > B > A$	$E > D > C > B > A$	$E > D > C > B > A$
6	$B > A$	$B > A$	$A \ ? \ B$	$A = B$
7	$C > B = D > A = E$	$B > C > A > D > E$	$A \ ? \ B \ ? \ C \ ? \ D \ ? \ E$	$A = B = C = D = E$
8	$E > D > C > B > A$	$A > B > C > D > E$	$A \ ? \ B \ ? \ C \ ? \ D \ ? \ E$	$E > D > C > B > A$

that a convergence only performance measures ignore diversity. If diversity is important to the user, these kind of metrics must be combined with diversity only ones in order to make a better evaluation in cases where the convergence is very similar for all sets, and they have dissimilar diversity.

6 Conclusions

In this study, we presented eight test cases for performance measures for multi-objective algorithms. Each test case consists on two or more non–dominated sets, constructed in such a way that it is evident what set is better than other. The test cases represent several topologies of the Pareto Front, and test the ability of the metrics to evaluate convergence and diversity. Also, the test cases challenges the robustness of the metrics in misleading cases. Four popular performance measures were tested. The result of the experiments corroborates the flaws in some metrics and the strength of others. All these experiments shows that a benchmark is helpful to decide what performance measures are better. The test cases presented here can be the basis to create bigger and better benchmarks. The test cases are available upon request.

References

1. Beume, N., Rudolph, G.: Faster s-metric calculation by considering dominated hypervolume as klee's measure problem. In: Kovalerchuk, B. (ed.) Computational Intelligence, pp. 233–238. IASTED/ACTA Press (2006)

2. Brockhoff, D., Zitzler, E.: Improving Hypervolume-based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods. In: Congress on Evolutionary Computation (CEC 2007) (2007)
3. Carlos, D.V., Coello, A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic/Plenum Publishers, New York (2002)
4. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms. John Wiley and Sons, Chichester (2001)
5. Fonseca, C.M., Paquete, L., López-Ibáñez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2006), pp. 1157–1163 (2006)
6. Hansen, M.P., Jaszkiewicz, A.: Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7 (1998)
7. Knowles, J., Corne, D.: On Metrics for Comparing Nondominated Sets. In: Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002), vol. 1, pp. 711–716 (2002)
8. Schott, J.: Fault tolerant design using single and multicriteria genetic algorithm optimization. Ph.D thesis, Massachusetts Institute of Technology (May 1995)
9. Zitzler, E.: Evolutionary Algorithms Multiobjective Optimization: Methods and Applications. Ph.D thesis, Swiss Federal Institute of Technology (ETH) (November 1999)
10. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 529–533 (2003)

Some Demonstrations about the Cardinality of Important Sets of Non–dominated Sets

Giovanni Lizárraga, Arturo Hernández, and Salvador Botello

Center for Research in Mathematics (CIMAT)
Department of Computer Science
A.P. 402, Guanajuato, Gto. 36240, México
giovanni@cimat.mx, artha@cimat.mx, botello@cimat.mx

Abstract. In multiobjective optimization, a very important element is the space of objective functions, usually called Z. The set of Ω of all non–dominated sets that we can generate with elements of Z is especially interesting, because it represent all possible output from an evolutionary multiobjective algorithm. In this study, we make some theoretical demonstrations about the cardinality of Ω and others important sets of non–dominated sets. After, we use these demonstrations to prove some theorems in the area of performance measures for evolutionary multiobjective algorithms.

1 Introduction

Evolutionary Multiobjective Optimization (EMO) consist on maximizing or minimizing (or a mixture) a vector of objective functions $F(x) = \langle f_1(x), f_2(x), \ldots, f_n(x)\rangle$ subject to constraints using Evolutive Algorithms (EA). The objective functions and constraints depend on a vector of variables $x \in \mathbf{R}^m$. We call X the set of all feasible vectors x, and Z the projection of X on objective functions space. Without loss of generality, we consider hereafter that we are minimizing the objective functions.

EMO uses the Pareto Optimality Criteria (POC). POC is defined through the relation between two vectors $z, w \in \mathbf{R}^n$ known as Pareto dominance. We have that z dominates w ($z \succeq w$) if $\forall i \in \{1, 2, \ldots, n\}, z^i \leq w^i \ \wedge \ \exists j \in \{1, 2, \ldots, d\} \mid z^j < w^j$, where v^k stands for the k–th component of vector v. The goal is to find a set of vectors known as the Pareto Set (PS) defined as $PS = \{x \in X \mid \forall y \in X, F(y) \not\succeq F(x)\}$. According to POC, all elements of PS are optimal, because they represent the different tradeoffs between the objective functions where it is not possible to improve one objective without degrading another. The projection of PS in objective functions space is called the Pareto Front (PF), and is usually described as a surface that represents the best tradeoffs possible between the objective functions.

In recent years, many multiobjective evolutionary algorithms (MOEAs) based on POC have been developed. For a review of these algorithms consult [2] and [1]. Instead of generating a single solution, these algorithms generate a finite set A of vector solutions $x \in X$ that approximate the PS. These approximation sets have

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 440–450, 2008.

the characteristic that $\forall x, y \in A, F(x) \not\succeq F(y) \ \wedge \ F(y) \not\succeq F(x)$ and are usually called non–dominated sets (NS). We can consider these NSs as approximations of the Pareto Set. Also, we can consider these NSs as approximations to the Pareto Front, if we locate the sets in their projection in objective function space. Hereafter in the article we must locate points, sets, vectors, and solutions in the space of the objective functions.

The set Z is very important because all solutions generated by a MOEA are subsets of Z. Z can have many topologies, it can be discrete, finite, continuous, etc. In the rest of the article we assume the most general case where $Z = R^n$.

Let define the set Ω as the set of all non–dominated sets we can generated with elements of Z. Ω is also a very interesting set, because all NSs generated by a MOEA are elements of Ω. One of the most important properties of a set is its cardinality. As we see later, the cardinality is related with the size of a set. The cardinality of Ω is especially important for performance measures, because many performance measures are functions whose domain is Ω (or a subset of Ω).

There are other sets of non–dominated sets that are important. Many MOEAs stores the non–dominated vector they have found using an archive of fixed size k. This means that, once the value of k is decided, we have that the set of all non–dominated sets we can generate are those non–dominated set of size less or equal to k. So, the set $\Omega_{\leq k}$ of all non–dominated sets of size less or equal to k is also interesting.

Finally, we want to introduce another important set. Define $\Omega_{\leq a}$ as the set of all non–dominated set of finite size. In practice, all MOEAs generates a finite number of explicit vector solution for a evolutionary multiobjective problem. So, $\Omega_{\leq a}$ is a better representation of all possible outputs of a MOEA than Ω.

We find the cardinality of the sets described above and use this information to explain some properties of the performance measures described in [5]. The rest of the article is organized as follows: in Section 2 we present some basic theorems from set theory, in Section 3 we use set theory to find the cardinality of some important sets. Later, in Section 4 we use the results of previous sections to make some demonstrations related to performance measures for multiobjective algorithms. Finally, in Section 5 we state our conclusions.

2 Set Theory

In this section we present an overview of set theory and present all the definitions and results that are necessary to demonstrate the theorems and lemmas in the following sections. The concepts in this section were taken from [3] and [4]. The demonstrations of the results presented here can be consulted in [3] and [4].

Two sets, A and B, are equivalent ($A \sim B$) if it is possible to make a correspondence between the elements of both sets in such a way that to every element of A correspond one and only one element of B; and to every element of B correspond one and only one element of A. This kind of correspondence is called a one–to–one correspondence. The equivalence property is reflexive, transitive and

symmetric. An injection from A to B is a one–to–one correspondence from A to a subset of B.

One of the most important concepts of set theory is that of cardinal numbers, or cardinality. Cardinal numbers are related with the size of a set. A cardinal number refers an arbitrary member of a family of mutually equivalent sets. For example, the cardinal number 4 represents any set equivalent to {1,2,3,4}, like {a,b,c,d}, {"dog", "rat", "cat", "mouse"}, etc. We represent the cardinal number of a set A by $|A|$, for example $|\{a, b\}| = 2$. Infinite sets also have cardinal numbers. For positive integers $\mathbf{N} = \{1, 2, 3, \ldots\}$, we represent its cardinal number by a. For real numbers $\mathbf{R}$ we represent $|\mathbf{R}|$ by c. We represent the cardinal number of the set of functions defined in a continuous interval by f.

It is not possible to make an injection from A to B if $|A| > |B|$ because there are not enough different elements in B to be associated with the elements in A. If $A \subset B$ then $|A| \leq |B|$. An interesting result from set theory is that it is impossible to make an injection from $\mathbf{R}$ to $\mathbf{N}$, the set of natural numbers is somehow "smaller" than the set of real numbers. For two infinite sets A and B, $|A| < |B|$ if and only if there is an injection from A to B but there is no injection from B to A. If we can make an injection from A to B, than $|A| \leq |B|$. It is proved that $a < c < f$.

A set A with cardinal number a is called countable and it is equivalent to the set of natural numbers. When listing its elements, a countable set is usually represented using "...", for example $A = \{a_1, a_2, \ldots\}$.

An interesting property of the infinite sets, like $\mathbf{N}$ and $\mathbf{R}$, is that it is possible to make a one–to–one correspondence between a infinite set and some of its subsets.

For two sets A and B, their union is represented by $A + B$. Their cartesian product is represented by $A \times B$. The cartesian product of a set with itself can be represented by an exponent. For example, $A \times A \times A = A^3$.

The power set of A, *P(A)*, is the set whose elements are all possible subsets of A and it is proved that $|A| < |P(A)|$. If $|A| = |B|$ then $|P(A)| = |P(B)|$.

The cardinal numbers of infinite sets are called transfinite numbers. The smallest transfinite number is a and all finite numbers are smaller than any transfinite number. We present a list of results of set theory, where $k > 0$ is a finite cardinal number and $m, n > 0$ are finite numbers.

a: $A \sim B \iff |A| = |B|$. Two sets are equivalent if and only if they have the same cardinal number.

b: $c+k = c+a = c+c = c$. For a set A of cardinality c, if we add a finite number of elements to A, the resulting set has cardinality c. The same occurs if we add countable many elements to A or if we add c many elements to A. For example $|\mathbf{R}+\{a, b, c\}| = |\mathbf{R}| = c$, $|(0, 1)+\mathbf{N}| = c$, $|[0, 1)+[1, 2)| = |[0, 1)| = c$. Similarly, we can extract a finite number of elements from a set of cardinality c and the resulting set has cardinality c.

c: $c \cdot k = c \cdot a = c \cdot c = c$. The cartesian product of a set A with cardinality c with a finite set results in a set with cardinality c. The same result is obtained

if the cartesian product is evaluated with a set of cardinality a or c. For example $|\mathbf{R} \times \{1,2,3\}| = c$, $|\mathbf{R} \times \mathbf{N}| = c$, $|\mathbf{R} \times \mathbf{R}| = c$.

d: $c^m = c^a = c$. A set A with cardinality c elevated to a finite exponent results in an equivalent set. For example, $|\mathbf{R}^5| = |\mathbf{R}| = c$. The same result is obtained if A is elevated to a.

e: Let $|A| = c$. $|A^c| = |P(A)|$. If we elevate a set A of cardinality c to the exponent c, the result is a set with a bigger cardinality. The same cardinality of the power set of A. For example, $\mathbf{R}^c \sim P(\mathbf{R})$.

f: $c^m \times c^n = c^{(m+n)} = c$. For example, $\mathbf{R}^5 \times \mathbf{R}^2 \sim \mathbf{R}^7 \sim \mathbf{R}$.

g: $c+c+\ldots = c$. The sum of countable many sets, each of them with cardinality c, results in a set of cardinality c. For example $[0,1) + [1,2) + \ldots + [k, k+1) + \ldots \sim \mathbf{R}$.

h: $|A| = |B| \Longrightarrow |P(A)| = |P(B)|$. If two sets have the same cardinality, then their power sets are equivalent.

i: $A \subset B \Longrightarrow |A| \leq |B|$. If A is a subset of B, then the cardinal number of A is less or equal to that of B.

j: Let $|A| = c$ and $|B| = a$: $f = |P(A)| > c = |P(B)| > a$.

k: If $C \subset B$ then: $A \sim C \Longrightarrow |A| \leq |B|$.

These results are used in the following sections.

3 The Cardinality of Some Important Sets of Non–dominated Sets

We introduce some demonstrations about the cardinality of the sets of non–dominated sets. These demonstrations are interesting by themselves from the theoretical point of view but they are also useful to prove the theorems we present later. We use extensively the Theorems (a)–(k) from Section 2.

First we present the line S. The line S is a structure we borrowed from [5] and we use it in the demonstration of some of the lemmas and theorems in this section. It is a segment of line in $\mathbf{R}^n$, aligned in a direction so the points in the line are a non–dominated set. An example of the line S is shown in Figure 1. The definition of S is the following:

Definition 1. *Choose $a, b \in \boldsymbol{R}$ with $a < b$. The line S is defined as $S = \{(z^1, z^2, \ldots, z^n) \in Z \mid z^1 \in (a,b), z^2 = b + a - z^1$ and $z^i = (a+b)/2\}$. Also, define Ω_S is the set of all non–dominated sets we can generate from S.*

Note that the definition of z^1 and z^2 implies that all points in S are non–dominated between them. For simplicity, we give a fixed value to the rest of the z^i. Ω_S is equivalent to the power set of S minus the empty set, because any subset of non–dominated set is also a non–dominated set with the exception of the empty set[1].

[1] In order to make the demonstrations shorter, we consider that the empty set is not a non–dominated set. This makes no difference because our demonstrations holds even if we consider that the empty set is a non–dominated set.

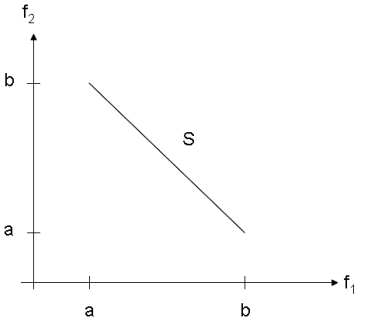


Fig. 1. A non–dominated set $S \subset \mathbf{R}^2$

In Section 1 we defined the set Ω of all non–dominated sets we can create from Z. We are interested in the following question. How many non–dominated sets can be created from Z? In other words, what is the cardinality of Ω? The answer is shown in the following theorem:

Theorem 1. *The cardinal number of Ω is f.*

Proof. Consider the line $S \subset Z$ described before. The argumentation is the following:

A: $\Omega_S \subset \Omega \subset P(\mathbf{R}^n)$, because of the definition of S, Ω and Ω_S.
B: $|\Omega_S| \leq |\Omega| \leq |P(\mathbf{R}^n)|$, because of (i) and A.
C: $\Omega_S = P(S) - \{\emptyset\}$, because the subset of a NS is also a NS.
D: $|\Omega_S| = f$, because the power set of a set with cardinality c has cardinality f (j), S has cardinality c and C.
E: $P(\mathbf{R}^n) \sim P(\mathbf{R})$, because of (d) and (h).
F: $|P(\mathbf{R}^n)| = f$, because of E, (a) and (j).
G: $f \leq |\Omega| \leq f$, because of B, D and F.
H: $|\Omega| = f$, this is a direct consequence of G. □

Other demonstrations that will result useful in the rest of this work are about the cardinality of the sets of non–dominated sets of a fixed size. For example, define Ω_k as the set of all non–dominated sets in $\mathbf{R}^n$ of size k, for $k > 0$. What is the cardinality of Ω_1, Ω_2 and in general Ω_k? We respond to these questions in the following three lemmas.

Lemma 1. *The cardinality of Ω_1 is c.*

Proof. By definition, Ω_1 is a set of the form $\{\{z\}|z \in Z\}$, so we can make a one to one correspondence between the elements $\{z\} \in \Omega_1$ with the corresponding $z \in Z$. This means that $\Omega_1 \sim Z$. The cardinality of Z is c, so recalling Theorem (a) from Section 2 we can conclude that $|\Omega_1| = c$. □

Lemma 2. *The cardinality of Ω_2 is c.*

Proof. Consider the line S described before. Choose a point $s_1 \in S$ and define $S' = S - \{s_1\}$. Define Ω'_2 as the set of all sets in the form $\{s_1, s\}$ for $s \in S'$. We have that all elements of Ω'_2 are non–dominated sets of size two.

A: $|S'| = c$, because if we take from a set of cardinality c a finite number of elements, the resulting set has cardinality c (b). We defined S' as $S - \{s_1\}$ and the cardinality of S is c, so the cardinal number of S' is c.
B: $|\Omega'_2| = c$, because we can make a one to one correspondence between Ω'_2 and S'. For this, associate each element $\{s_1, s\} \in \Omega'_2$ with the corresponding element $s \in S'$. But the cardinality of S' is c (A), so the cardinal number of Ω'_2 is c (a).
C: $\Omega'_2 \subset \Omega_2$, because the elements of Ω'_2 are non–dominated sets of size 2 and Ω_2 is the set of all non–dominated sets of size 2.
D: $|\Omega'_2| \leq |\Omega_2|$, because Ω'_2 is a subset of Ω_2 (C), so its cardinality must be less or equal to that of Ω_2 (i).
E: $c \leq |\Omega_2|$, because the cardinal number of Ω'_2 is less or equal to the cardinal number of Ω_2 (D), and the cardinal number of Ω'_2 is c (B).
F: $|\Omega_2| \leq |\mathbf{R}^{2n}|$, because of (k) and because we can make an injection from Ω_2 to $\mathbf{R}^{2n}$. For this, sort the vectors $v =< v^1, v^2, ...v^n > \in A$ for $A \in \Omega_2$ using the following rule: v precedes u, for $v, u \in A$, if $v^1 < u^1$ or if $v^r < u^r$ when $v^i = u^i$ for $1 \leq i \leq r-1$. This way, every non–dominated set $A \in \Omega_2$ is associated with an unique pair of ordered vectors $v_1 =< v_1^1, v_1^2, \ldots, v_1^n >, v_2 =< v_2^1, v_2^2, \ldots, v_2^n >$. Associate each non–dominated set $A \in \Omega_2$ with the point $(v_1^1, v_1^2, \ldots, v_1^n, v_2^1, v_2^2, \ldots, v_2^n) \in \mathbf{R}^{2n}$ and we have the desired injection.
G: $|\Omega_2| \leq c$, because the cardinal number of Ω_2 is less or equal to the cardinal number of $\mathbf{R}^{2n}$ (F) and the cardinal number of $\mathbf{R}^{2n}$ is c (d).
H: $c \leq |\Omega_2| \leq c$, because of E and G.
I: $|\Omega_2| = c$, this is a direct consequence of H. □

The proof of Lemma 2 can be extended for Ω_k for $k > 2$, as it is shown in the following demonstration.

Lemma 3. *The cardinality of Ω_k, for $k > 2$, is c.*

Proof. Consider the line S described before. Choose $k-1$ different points $s_1, s_2, \ldots, s_{k-1} \in S$ and define $S' = S - \{s_1, s_2, \ldots, s_{k-1}\}$. Define Ω'_k as the set of all sets in the form $\{s_1, s_2, \ldots, s_{k-1}, s\}$ for $s \in S'$. We have that all elements of Ω'_k are non–dominated sets of size k.

A: $|S'| = c$, because if we take from a set of cardinality c a finite number of elements, the resulting set has cardinality c (b). We defined S' as $S - \{s_1, s_2, \ldots, s_{k-1}\}$ and the cardinality of S is c, so the cardinal number of S' is c.
B: $|\Omega'_k| = c$, because we can make a one–to–one correspondence between Ω'_k and S'. For this, associate each element $\{s_1, s_2, \ldots, s_{k-1}, s\} \in \Omega'_k$ with the corresponding element $s \in S'$. But the cardinality of S' is c (A), so the cardinal number of Ω'_k is c (a).

C: $\Omega'_k \subset \Omega_k$, because the elements of Ω'_k are non–dominated sets of size k and Ω_k is the set of all non–dominated sets of size k.
D: $|\Omega'_k| \leq |\Omega_k|$, because Ω'_k is a subset of Ω_k (C), so its cardinality must be less or equal to that of Ω_k (i).
E: $c \leq |\Omega_k|$, because the cardinal number of Ω'_k is less or equal to the cardinal number of Ω_k (D), and the cardinal number of Ω'_k is c (B).
F: $|\Omega_k| \leq |R^{kn}|$, because of (k) and because we can make an injection from Ω_k to R^{kn}. For this, sort the vectors $v =< v^1, v^2, ...v^n > \in A$ for $A \in \Omega_k$ using the following rule: v precedes u, for $v, u \in A$, if $v^1 < u^1$ or if $v^r < u^r$ when $v^i = u^i$ for $1 \leq i \leq r-1$. This way, every non–dominated set $A \in \Omega_k$ is associated with an unique list of ordered vectors $v_1 =< v_1^1, v_1^2, \ldots, v_1^n >, v_2 =< v_2^1, v_2^2, \ldots, v_2^n >, \ldots, v_k =< v_k^1, v_k^2, \ldots, v_k^n >$. Associate each non–dominated set $A \in \Omega_k$ with the point $(v_1^1, v_1^2, \ldots, v_1^n, v_2^1, v_2^2, \ldots, v_2^n, \ldots, v_k^1, v_k^2, \ldots, v_k^n) \in R^{kn}$ and we have the desired injection.
G: $|\Omega_k| \leq c$, because the cardinal number of Ω_k is less or equal to the cardinal number of R^{kn} (F) and the cardinal number of R^{kn} is c (f).
H: $c \leq |\Omega_k| \leq c$, because of E and G.
I: $|\Omega_k| = c$, this is a direct consequence of H. □

So, based on the demonstrations presented before, we introduce the following theorem:

Theorem 2. *The cardinal number of Ω_k, where k is a positive integer, is c.*

Theorem 2 is useful in the demonstration of the two following theorems. These theorems are related to the cardinality of $\Omega_{\leq k}$ and $\Omega_{<a}$.

Theorem 3. *The cardinality of $\Omega_{\leq k}$ is c.*

Proof. We can represent $\Omega_{\leq k}$ with the following sum:

$$\Omega_{\leq k} = \Omega_1 + \Omega_2 + \ldots + \Omega_k \ . \tag{1}$$

the argumentation is as follows.

A: $|\Omega_k| = c$, because of Theorem 2.
B: $|\Omega_{\leq k}| = |\Omega_1 + \Omega_2 + \ldots + \Omega_k|$, because of (1).
C: $|\Omega_{\leq k}| = |\Omega_1| + |\Omega_2| + \ldots + |\Omega_k|$, because the different Ω_i are mutually disjoint.
D: $|\Omega_{\leq k}| = k \cdot c$, because of A and C.
E: $|\Omega_{\leq k}| = c$, because of D and because the product of c with a natural number $k > 0$ is equal to c (c). □

Theorem 4. *The cardinal number of $\Omega_{<a}$ is c.*

Proof. We can represent $\Omega_{<a}$ with the following sum:

$$\Omega_{<a} = \Omega_1 + \Omega_2 + \Omega_3 + \ldots \ . \tag{2}$$

Note that this sum have countable many elements, because we can make a one–to–one correspondence between the positive integers and the elements of the sum. For this we only need to associate each positive integer k with the corresponding Ω_k. The values of k increase without a limit, but they are always finite.

A: $|\Omega_k| = c$, because of Theorem 2.
B: $|\Omega_{\leq a}| = |\Omega_1 + \Omega_2 + \Omega_3 + \ldots|$, because of (2).
C: $|\Omega_{\leq a}| = |\Omega_1| + |\Omega_2| + |\Omega_3| + \ldots$, because the different Ω_k are mutually disjoint and B.
D: $|\Omega_{\leq a}| = c + c + c + \ldots$, because of A and C.
E: $|\Omega_{\leq a}| = c$, because of D and (g). □

4 Cardinalities and Performance Measures

In this section we make an extensive use of the Theorems 1– 4 from Section 3 and Theorems (a)–(k) from Section 2.

An open problem in multiobjective optimization is how to evaluate the performance of different MOEAs. After every run, a MOEA generates a non–dominated set that approximates the Pareto Front. So, the evaluation of the MOEAS can be reduced to evaluate the NSs it generates. A popular method to evaluate a non–dominated set is to use a unary quality indicator [5]. Define Ω_r as the set of all non–dominated sets that can be generated by a MOEA, a unary quality indicator is a function $I : \Omega_r \rightarrow R$, that takes one NS as an argument (hence it is named unary) and returns a real number as an output. This real number is a measure of how good a NS is, and it is used to compare different NSs. For example, if we consider that the bigger the value of I the better, for two NSs A and B, if $I(A) > I(B)$ we can conclude that A is better than B. It is also valid to use several unary quality indicators in order to evaluated two non–dominated sets.

In [5] it was also defined the concept of unary comparison method. A unary comparison method $UC_{I,E}$ is formed by a combination $\mathbf{I} = (I_1, I_2, \ldots, I_k)$ of k unary quality indicators and a function $E : \mathbf{R}^k \times \mathbf{R}^k \rightarrow \{True, False\}$. For two NSs A and B, if $UC_{I,E}(A, B)$ is true, then the comparison method considers that A is better than B. If $UC_{I,E}(A, B)$ is false, then the comparison method considers that A is not better than B.

In [5] it was introduced a property for comparison methods known as compatibility and completeness. It is considered that compatibility and completeness property is very important to have, because a compatible and complete comparison method is able to decide whether a NS A is better than a NS B. It was proved [5] that a unary comparison method can not be compatible and complete. This is described in the following theorem:

Theorem 5. *For multiobjective problems with 2 or more objectives, there exists no unary comparison method with a finite number k of performance measures in* $\boldsymbol{I}$*, that is compatible and complete.*

Theorem 5 has very important implications. It means that unary metrics have a limited capacity to evaluate whether a non–dominated set is better than another. In order to demonstrate this theorem, we need Lemma 4.

Lemma 4. *Let* $Z = \{(z^1, z^2, \ldots, z^n) \in \boldsymbol{R}^n \mid a < z^i < b, 1 < i < n\}$, *be an open hypercube in* $\boldsymbol{R}^n$ *with* $n \geq 2$, $a, b \in \boldsymbol{R}$. *If there exist a compatible and complete unary comparison method with* $\boldsymbol{I} = (I_1, I_2, ..., I_k)$, *and an interpretation function* E, *then for all* $A, B \in \Omega$ *with* $A \neq B$ *there is at least one* I_j *in* $\boldsymbol{I}$ *such that* $I_j(A) \neq I_j(B)$.

In other words, for a compatible and complete unary comparison method (CCUC), if $A \neq B$ then $\mathbf{I}(A) \neq \mathbf{I}(B)$. Define Υ as the set of all different vectors $\mathbf{I}$ we can generate with k quality indicators. The cardinal number of Υ is c, because $\Upsilon = \mathbf{R}^k \sim \mathbf{R}$, (see (d) in Section 2). Due to Lemma 4 there must be an injection from Ω to $\mathbf{R}$. The demonstration of Lemma 4 can be found in [5]. Next, we present a demonstration of Theorem 5 that is equivalent to the demonstration presented in [5].

Proof (Theorem 5). Let $Z = R^n$ and $\Omega_r = \Omega$.

A: $|\Omega_r| = f$, see Theorem 1.
B: $|\Upsilon| = c$, because of the definition of Υ and (d).
C: $c < f$, because of (j).
D: We need to make an injection from Ω_r to Υ, because of Lemma 4.
E: It is impossible to make an injection from Ω_r to Υ, because of A, B and C. □

The conclusion is that no comparison method based on a finite number of unary performance measure can be compatible and complete, because it leads to an absurd result. The central part of the demonstration is that we can not make an injection from Ω_r to Υ. This part of the demonstration is central for the rest of the study in this work, so we refer to it as the "cardinality contradiction".

Ziztler et al. [5] describe two conditions in which it is possible to construct a compatible and complete unary comparison method. The first condition is the use an infinite number of unary quality indicators. They even mention the empirical Attainment Function as a basis to construct a CCUC using a infinite number of quality indicators. An interesting question is why it is possible to construct such CCUC under the conditions just mentioned. The reason is very simple, and it is described in the following corollary.

Corollary 1. *If we use c–many unary quality indicators, the cardinality contradiction vanishes.*

Proof. Redefine $\mathbf{I}$ as a combination of c–many unary performance measures. Let $Z = R^n$ and $\Omega_r = \Omega$.

A: $|\Omega_r| = f$, because of Theorem 1.
B: $|\Upsilon| = f$, because the number of different combinations of unary performance measures we can generate from $\mathbf{I}$ is $\mathbf{R}^c = f$ (e).
C: We need to make an injection from Ω_r to Υ, because of Lemma 4.
D: It is possible to make an injection from Ω_r to Υ, because of A and B. □

So, the cardinality contradiction vanishes if we use c–many unary quality indicators. Note that this is not true if we redefine $\mathbf{I}$ to contain a–many unary quality indicators, because in this case $|\Upsilon| = |\mathbf{R}^a| = c < f = |\Omega|$, so the cardinality contradiction holds.

The second special condition introduced in [5] is that there exist a comparison method that is compatible and complete if the size of the NSs to compare is equal or less than a fixed value k. They even described that comparison method. Again, an interesting question is why it is possible to construct such a comparison method with the new conditions. Just like in the previous case, new conditions leads us to a new result. We state this in the following corollary:

Corollary 2. *If we restrict the size of the non–dominated sets under comparison to be smaller or equal than a fixed value k the cardinality contradiction vanishes.*

Proof. Let $Z = R^n$ and $\Omega_r = \Omega_{\leq k}$.

A: $|\Omega_r| = c$, because of Theorem 3.
B: $|\Upsilon| = c$, because the number of different combinations of unary performance measures we can generate from I with k unary metrics is $|\mathbf{R}^k| = c$ (d).
C: We need to make an injection from Ω_r to Υ, because of Lemma 4.
D: It is possible to make an injection from Ω_r to Υ, because of A and B. □

Unfortunately, the comparison method mentioned above is not useful in practice [5]. However, considering Corollary 2, we wonder if other more useful comparison methods can be created.

Finally, we introduced a third special condition where the cardinality contradiction vanish. The new special condition is to consider only non–dominated sets of arbitrary finite size. Note that this condition is less restrictive than the one mentioned in Corollary 2, because we do not need to fix the size of the NSs to compare.

Corollary 3. *If we only compare non–dominated sets $A \in \Omega_{<a}$, the cardinality contradiction vanishes.*

Proof. Let $Z = R^n$ and $\Omega_r = \Omega_{\leq a}$

A: $|\Omega_r| = c$, because of Theorem 4.
B: $|\Upsilon| = c$, because of the definition of Υ and (d).
C: $\Omega_r \sim \Upsilon$, because of A, B and (a).
D: We need to make and injection from Ω_r to Υ, because of Lemma 4.
E: It is possible to make an injection from Ω_r to Υ, because of C. □

The consequence of Corollary 3 is that we can not affirm that a compatible and complete comparison method based on a finite number of unary metrics is impossible to construct. All evolutionary multiobjective algorithms known so far generate a finite number of vectors as an output. An algorithm that generates an infinite number of explicit vector solutions, in finite time, is not possible. So, we have that the conditions of Corollary 3 are met in practice.

5 Conclusions

We presented an analysis of the cardinality of different sets of non–dominated sets. The theorems proved in this work are useful for the theoretical development of Multiobjective Evolutionary Computation theory. Using the results of our analysis, we made some interesting demonstrations in the area of performance measures for MOEAs. Besides, we proved that under practical conditions we can not affirm that a compatible and complete unary comparison method is impossible to construct. Actually, we can neither affirm the opposite. For the moment, it is unknown to us if such comparison method can be created or not.

References

1. Carlos, D.V., Coello, A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic/Plenum Publishers, New York (2002)
2. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms. John Wiley and Sons, Chichester (2001)
3. Fraenkel, A.A.: Set Theory and Logic. Addison–Wesley Publishing Company, Reading (1966)
4. Kamke, E.: Theory of Sets. Dover Publications, Inc., New York (1950)
5. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 529–533 (2003)

Looking Inside Particle Swarm Optimization in Constrained Search Spaces

Jorge Isacc Flores-Mendoza and Efrén Mezura-Montes

Laboratorio Nacional de Informática Avanzada (LANIA A.C.)
Rébsamen 80, Centro, Xalapa, Veracruz, 91000, México
jflores@lania.edu.mx, emezura@lania.mx

Abstract. In this paper, the behavior of different Particle Swarm Optimization (PSO) variants is analyzed when solving a set of well-known numerical constrained optimization problems. After identifying the most competitive one, some improvements are proposed to this variant regarding the parameter control and the constraint-handling mechanism. Furthermore, the on-line behavior of the improved PSO and some of the most competitive original variants are studied. Two performance measures are used to analyze the capabilities of each PSO to generate feasible solutions and to improve feasible solutions previously found i.e. how able is to move inside the feasible region of the search space. Finally, the performance of this improved PSO is compared against state-of-the-art PSO-based algorithms. Some conclusions regarding the behavior of PSO in constrained search spaces and the improved results presented by the modified PSO are given and the future work is established.

1 Introduction

The use of nature-inspired heuristics to solve complex search problems, like numerical optimization problems, has been extended in recent years. Evolutionary Computing (EC) [1], which emulates natural selection and survival of the fittest comprises the first set of these kind of techniques. However, in the mid 1990's the emulation of emergent social behaviors among insects and bird flocks has been proposed as a new area known as Swarm Intelligence (SI) [2]. The two SI initial paradigms are Ant Colony Optimization (ACO) [3] and Particle Swarm Optimization [2]. ACO is based on the foraging behavior of ants and has been mainly used to solve combinatorial optimization problems. On the other hand, PSO is based on the social behavior of bird flocks when moving from one place to another and was proposed mainly to solve numerical optimization problems.

PSO is based on social relationships established among simple individuals in a group. There is a leader in the flock which is followed by the others members. However, each member has a memory about its position in the group. In this way, the individual is able to take a decision based on its own knowledge (cognitive element) and also based on the behavior of its neighbors (social element). PSO is easy to implement and its performance is sometimes better than other nature-inspired heuristics [2].

In PSO, an initial swarm of solutions called particles $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]$ are generated at random. These particles will "fly" in the search space as to locate promising areas and reach a good (optimal) solution. Particles find their search direction

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 451–461, 2008.

by combining social (the position of the best particle in the swarm called "leader") and cognitive (its best position reached so far, called "pbest") information. The position of a particle is changed by adding an updated velocity to the current position: $\boldsymbol{x}(t+1) = \boldsymbol{x}(t) + \boldsymbol{v}(t+1)$, called flight formula. Two different approaches to calculate the velocity vector are presented in this study. The first is the PSO with inertia weight and the second is the PSO with constriction factor:

PSO with inertia weight. The inertia weight [2] was added to the velocity update formula to calibrate the influence of the previous particle's velocity $\boldsymbol{v}_i(t)$. The formula to calculate the new velocity $\boldsymbol{v}_i(t+1)$ is the following: $\boldsymbol{v}_i(t+1) = w * \boldsymbol{v}_i(t) + c_1 * rand() * (\boldsymbol{x}_{pbest_i} - \boldsymbol{x}_i) + c_2 * rand() * (\boldsymbol{x}_{gBest_i} - \boldsymbol{x}_i)$, where w is the inertia weight, $\boldsymbol{x}_{pbest_i}$ is the particle's pbest, $\boldsymbol{x}_{gBest_i}$ is the position of the leader, c_1 and c_2 are the acceleration constants (user-defined) which control the influence of the cognitive (memory of the particle) and social (position of the leader) elements respectively, $rand()$ is a function that generates a uniform-distributed random real number between 0 and 1.

PSO with constriction factor. It was proposed by Clerc and Kennedy in [4] in order to improve the exploration-exploitation capabilities of PSO. The constriction factor k is included in the velocity update formula as follows: $\boldsymbol{v}_i(t+1) = k * [\boldsymbol{v}_i(t) + c_1 * rand() * (\boldsymbol{x}_{pbest_i} - \boldsymbol{x}_i) + c_2 * rand() * (\boldsymbol{x}_{gBest_i} - \boldsymbol{x}_i)]$, where the constriction factor k is calculated by means of the acceleration constants c_1 and c_2 (the remaining elements of the formula are the same of the inertia weight velocity update formula). The authors claim that the constriction factor PSO variant, under certain parameters offers a better velocity control [4]. However, these conclusions are based on unconstrained numerical optimization problems.

There are two main communication variants in PSO: (1) Global best (GBPSO), where all particles can communicate in the swarm (star social network structure) and there is just one global leader ($gBest$), and (2) Local best (LBPSO), where particles can only communicate with others in their vicinities (ring social network structure) and there are several leaders ($lBest_i$) depending of the number of neighborhoods defined. Usually, GBPSO converges faster than LBPSO, but the first has a higher tendency to get trapped in local optima while the second may be more robust to avoid them [2]. The general PSO pseudocode is presented in Figure 1.

The problem of interest in this paper is the numerical constrained optimization problem (NCOP), which can be defined as follows: Find $\boldsymbol{x}$ which minimizes $f(\boldsymbol{x})$ subject to: $g_i(\boldsymbol{x}) \leq 0, \quad i = 1, \ldots, m$, and $h_j(\boldsymbol{x}) = 0, \quad j = 1, \ldots, q$ where $\boldsymbol{x} \in \mathbb{R}^n$ is the

- Begin
  - Generate a swarm of random solutions.
  - Evaluate the fitness of each particle in the swarm
  - do
    - Select the leader(s) of the swarm.
    - For each particle, update its position with the flight formula.
    - Evaluate the fitness of the new position of each particle
    - Update the $pbest$ (memory) value of each particle.
  - Until $t = Gmax$.
- End

Fig. 1. Basic PSO algorithm. GMAX is the number of generations of the algorithm.

vector of solutions $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$, where each $x_i, \;\; i = 1, \ldots, n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$ which define the search space $\mathcal{S}$, $\mathcal{F}$ is the feasible region and $\mathcal{F} \subseteq \mathcal{S}$; m is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or nonlinear). Equality constraints are transformed into inequalities constraints of the form: $\mid h_j(\boldsymbol{x}) \mid -\epsilon \leq 0$, where ϵ is the tolerance allowed (a very small value).

PSO, as other bio-inspired heuristics like EC, in their original versions, are designed to solve unconstrained optimization problems i.e. they lack a mechanism to incorporate feasibility information of solutions in their fitness values. There are excellent surveys about the research about constraint-handling techniques used in EC and also in SI [5]. The most popular approach to deal with constraints is the use of penalty functions, which aim is to decrease the fitness (i.e. quality measure) of infeasible solutions. There are other methods based on preserving feasible solutions, methods which make a clear distinction between feasible and infeasible solutions and also hybrid methods [5].

Regarding the use of PSO to solve NCOPs there are works reported in the specialized literature. They can be classified in (1) approaches based on penalty function and (2) based on methods of separation of objective and constraints. Examples of methods based on penalty functions are the following: Parsopoulos and Vrahatis [6] used a static penalty function and stochastic parameter in a combined GBPSO variant with constriction factor. Li, Tian and Kong [7] proposed a GBPSO with inertia weight using an adaptive penalty function and a mutation strategy based on the population diversity. Krohling and Do Santos Coehlo [8] used a co-evolutionary approach in a Lagrangian function with two sub-swarms, one of them optimizes the original problem and the other one aims to find the optimal values for the Lagrangian values. He, Prempain and Wu [9] proposed a GBPSO with inertia weight and with a "fly-back" (i.e. death penalty) mechanism which only lets the PSO fly inside the feasible region of the search space. On the other hand, approaches based on separation of objective function and constraints are the following: Toscano and Coello [10] proposed a turbulence operator in a GBPSO with inertia weight where the leader is chosen based on the lowest number of violated constraints (when infeasible particles are in the current swarm). Liang and Suganthan [11] used a LBPSO where each sub swarm (i.e. neighborhood) is focused on either satisfying a single constraint or optimizing the objective function, regardless of feasibility information. Cagnina et al. [12] presented a combination of global-local best PSO with inertia weight and a constraint handling method based on feasibility rules which prefer feasible solutions over infeasible ones and among infeasible solutions chooses those with a lower amount of constraint violation. Lu and Chen [13] proposed the use of a bi-objective approach based on a GBPSO with inertia weight, one objective is the original objective function and the second one is the sum of constraint violation.

As it can be seen from the related work, the design is centered on the constraint-handling technique and, most of the time, the PSO variant is chosen without knowing if, indeed, is the most suitable for constrained search spaces and for the constraint-handling technique used. Consequently, it is common to see additional operators like mutation [10], combination of communication models e.g. global-local best PSO [12,6] and mechanisms like co-evolution [8]. In this work, a performance analysis of original PSO variants when solving NCOPs is presented. A simple parameterless constraint-handling

technique was used as to do not introduce any additional bias to the original PSO. Based on the most competitive variant, two small improvements are made to it regarding parameter control and constraint-handling. After that, the on-line behavior of this improved variant is compared with respect to the original ones by using two performance measures. The hypothesis of this work is that the more knowledge about the original PSO behavior may lead to design competitive but less complex approaches, in this case, to solve NCOPs by using the search abilities provided by the search engine itself.

The paper is organized as follows: Section 2 presents an empirical comparison of four basic PSO variants. Based on the results obtained, the most competitive variant is improved in Section 3. The on-line behavior analysis of this new approach, as well as other PSO variants, is presented in Section 4. Finally, in Section 5 the conclusions of this work are enumerated and the future work is established.

2 Empirical Comparison of PSO Variants

As a first step in this proposal, the four most used PSO variants in constrained optimization, in their original versions, were compared in a benchmark found in the specialized literature [13] which comprises minimization problems with different features (See Table 1). Each PSO variant used in this comparison was implemented.

The goal of this experiment is to detect which original variant (without additional operators and complex mechanisms) is able to solve this set of test problems [13] with the best performance. The four variants chosen were: (1) GBPSO with inertia weight, (2) LBPSO with inertia weight, (3) GBPSO with constriction factor and (4) LBPSO with constriction factor.

Table 1. Main features for each benchmark problem used in the experiments. ρ is the estimated size of the feasible region with respect to the whole search space [5], n is the number of decision variables, LI is the number of linear inequality constraints, NI the number of nonlinear inequality constraints, LE is the number of linear equality constraints and NE is the number of nonlinear equality constraints.

Problem	n	Type of function	ρ	LI	NI	LE	NE
g01	13	quadratic	0.0003%	9	0	0	0
g02	20	nonlinear	99.9973%	0	2	0	0
g03	10	nonlinear	0.0026%	0	0	0	1
g04	5	quadratic	27.0079%	0	6	0	0
g05	4	nonlinear	0.0000%	2	0	0	3
g06	2	nonlinear	0.0057%	0	2	0	0
g07	10	quadratic	0.0000%	3	5	0	0
g08	2	nonlinear	0.8581%	0	2	0	0
g09	7	nonlinear	0.5199%	0	4	0	0
g10	8	linear	0.0020%	3	3	0	0
g11	2	quadratic	0.0973%	0	0	0	1
g12	3	quadratic	4.7697%	0	1	0	0
g13	5	nonlinear	0.0000%	0	0	0	3

The constraint-handling mechanism used in this work was originally proposed by Deb [14] and it consists on three feasibility rules: (1) Between 2 feasible solutions, the one with the highest fitness value wins, (2) if one solution is feasible and the other one is infeasible, the feasible solution wins and (3) if both solutions are infeasible, the one with the lowest sum of normalized constraint violation is preferred $\left(\sum_{i=1}^{m+q} max(0, g_i(\boldsymbol{x}))\right)$.

The parameter values for each PSO variant were defined as follows: 80 particles and 2000 generations (160,000 evaluations), $c_1 = 2.7$ and $c_2 = 2.5$ for all PSO variants, for the two local best variants we used 8 neighborhoods, $w = 0.7$ for both inertia weight variants and $k = 0.729$ [8] for both constriction factor variants. Equality constraints tolerance was set to ϵ =0.0001. These values were chosen empirically (by using set of independent runs per different parameter values as to find better statistical results for all variants) favoring the best performance of the PSO variants. 30 independent runs per variant per problem were performed. The statistical results are summarized in Table 2, first six columns. The best (B), mean (M) and standard deviation (SD) values are reported. Bks is the best known solution per test problem. Quality of a solution is measured by the B result, while consistency is measured by a better M and/or lower SD values. From those results, the variant Local best with constriction factor was clearly the most competitive (better statistical values on 11 test problems and similar good results just in 2 test problems). Furthermore, it was the only variant which reached the feasible region in all 13 problems. Global best variants presented premature convergence and failed to generate feasible solutions in two problems (g05 and g13). Finally, the LBPSO with inertia weight also presented convergence to local optima and failed to reach the feasible region in problem g13.

3 Proposed Algorithm: Modified PSO (MPSO)

The results obtained in the previous empirical comparison showed that the LBPSO with constriction factor was the most competitive variant. Two collateral conclusions from the previous study were that (1) LBPSO performs better in constrained search spaces and (2) as other nature-inspired heuristics, PSO is very sensitive to its parameter values (it was quite difficult to fine-tune them). Therefore, a parameter control mechanism was added to the most competitive variant. Two parameters have a strong influence in the velocity update formula used in the LBPSO with constriction factor: The constriction factor k and the acceleration constant c_2 (related with the social element in the swarm i.e. the position of the leader). Thus, as to promote different behaviors in the particles of the swarm, regardless of the neighborhood they belong, a dynamic parameter control for these two parameters is proposed for a subset of particles in the swarm. Hence, at each generation, some particles will use the k and c_2 static values and the remaining ones will use dynamic values, which will be ascending as the generations go on, until these values are the same of the static ones. The expected effect is that some particles will move at a pre-defined ratio (based on the static values) and others will move slowly at the beginning and faster as the process advances. The proposed function to dynamically adapt the values is the following: $f(y) = y^4$, where y is the number of

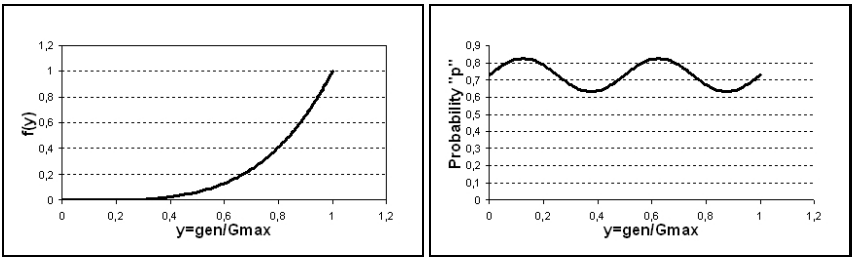


Fig. 2. Left: Behavior of the dynamic function to adapt the k' and c_2' values. y is the number of the current generation divided by $Gmax$. This behavior promotes some particles to move slowly at the beginning and quickly increasing its velocity as to equal the velocities of the remaining particles. Right: Behavior of the dynamic adaptation for the percentage of particles that uses the increasing parameter values.

current generation divided by $Gmax$ of generations. The expressions to modify k' and c_2' (dynamic values) are: $k' = k * f(y)$ and $c_2' = c_2 * f(y)$ (see left graph on Figure 2).

The percentage of particles that will use the dynamic values is also dynamic. An oscillatory effect provided a better performance (based on results of previous set of runs with different effects). It follows that the rate of particles that will use the dynamic values will vary between 60% and 80% during a single run based on a probability p calculated as follows: $p = k + \frac{(\sin(4.0 \times \pi \times y))}{10.3}$, where y is defined as mentioned before and k is the fixed value for the constriction factor (see right graph in Figure 2). In this modified PSO, constraints are handled with the same constraint mechanism used in the empirical comparison presented in Section 2. However, the sum of constraint violation was calculated separately for the equality and inequality constraints. Therefore, Pareto dominance [15] is used to select the best between two infeasible solutions (criterion 3). MPSO pseudocode is shown in Figure 3.

4 On-Line Behavior

To analyze more in-depth the behavior of this modified PSO (MPSO), two performance measures were used. The first one is the number of solution evaluations required to generate the first feasible solution (EVALS) [16]. A lower value is preferred because it means a low computational cost to reach the feasible region of the search space. The second one measures the capacity of an algorithm to improve the first feasible solution i.e. it measures the ability of an algorithm to move inside the feasible region of the search space. The formula, called PROGRESS RATIO and proposed in [16] is the following: $P = |\ln \sqrt{\frac{f_{min}(G_{ff})}{f_{min}(GMAX)}}|$, where $f_{min}(G_{ff})$ is the value of the objective function of the first feasible solution found and $f_{min}(GMAX)$ is the value of the objective function of the best feasible solution at the end of the process. A higher value indicates a better improvement inside the feasible region. The MPSO is compared with respect to two of the original variants used in the first experiment: GBPSO and LBPSO, both with constriction factor. The parameters used are the same reported in the first

- Begin
  - Generate a swarm of random solutions.
  - Evaluate the fitness of each particle in the swarm.
  - do
    - Select the leader(s) of the swarm **by using the modified feasibility rules**
    - For each particle, update its position with the flight formula **A subset of the swarm uses increasing parameters**
    - Evaluate the fitness of the new position of each particle.
    - Update the $pbest$ (memory) value of each particle **by using the modified feasibility rules**.
  - Until $t = Gmax$
- Stop

Fig. 3. Modified PSO, the modifications are remarked in **boldface**. GMAX is the number of generations of the algorithm.

experiment. The only differences with respect to the MPSO is that k and c_2 are dynamically controlled as well as the percentage of particles using these dynamic values. 30 independent runs per PSO per problem per performance measure were performed. Statistical values (best, mean and standard deviation) are summarized in Table 2 (last six columns). In order to get more statistical support to the results, nonparametric statistical tests (Kruskal-Wallis and Mann-Whitney) were computed for each sample compared as to verify that the differences shown in the samples are significant (95%-level of confidence). The results of these tests showed that the differences are significant in all cases except in problems g08, g11 and g12 for EVALS and in problems g04, g08, g12 and g13 for PROGRESS RATIO.

EVALS. The GBPSO with constriction factor quickly locates the feasible region in problems g01, g06, g07 and g10, but in g13 it fails to find it. The MPSO reaches faster the feasible region in problems g03, g05 and g13. The LBPSO with constriction factor (the base variant for the MPSO) did not provide better results in any function. In the remaining problems, the behavior was very similar in the three approaches compared. The results suggest that the GBPSO is able to generate feasible solutions faster in problems with a combination of linear and nonlinear inequality constraints which define a very small feasible region with respect to the search space (problems g01, g06, g07 and g10). On the other hand, the MPSO was faster in problems with only equality constraints (g03, slightly better in g13) and problems with a combination of inequality and equality constraints (g05).

PROGRESS RATIO. The MPSO could obtain a better improvement (based on quality and consistency) inside the feasible region in problem g01. Furthermore, MPSO provided the best quality improvement in problems g09 and g10. The LBPSO with constriction factor was clearly better in problem g06 and it was more robust in problem g10. The GBPSO with constriction factor did not provide a better result in any test problem. The above results show that MPSO has slightly better capabilities to maintain a progress inside the feasible region, regardless the features of the problem being solved in this benchmark. The dynamic parameter adaptation may be useful as to promote a better exploration of the search space, and mostly of the feasible region.

Based on the overall obtained results, the GBPSO with constriction factor is faster to reach the feasible region, but fails in some problems. However, MPSO is more

Table 2. First section: Statistical results of 30 independent runs on the 13 test problems for the four PSO variants compared. Second section: Statistics of EVALS and PROGRESS RATIO on 30 independent runs on the same 13 test problems. "(n)" means that in "n" runs, out of 30, the feasible region was found. "-" means that no feasible solutions were found in any single run. In **boldface** the best obtained result is remarked.

COMPARISON OF 4 PSO VARIANTS						PERFORMANCE MEASURES					
						EVALS			PROGRESS RATIO		
Problem		global best (w=0.7)	global best (k=0.729)	local best (w=0.7)	local best (k=0.729)	global best (k)	local best (k)	MPSO	global best (k)	local best (k)	MPSO
g01	B	-14.961	-14.951	-14.999	**-15.000**	**162**	246	252	0.302	0.346	**0.368**
	M	-11.217	-11.947	-12.100	**-13.363**	**306**	368	419	0.196	0.266	**0.295**
	SD	2.482	1.813	3.055	**1.397**	64.096	**54.107**	77.781	0.059	0.050	**0.038**
g02	B	-0.655973	-0.634737	-0.614785	**-0.790982**	0	0	0	**1.388**	1.373	1.218
	M	-0.606774	-0.559591	-0.543933	**-0.707470**	0	0	0	0.884	**1.015**	1.013
	SD	0.026463	0.030346	**0.020048**	0.059237	0.000	0.000	0.000	0.123	0.107	**0.099**
g03	B	-0.080	-0.019	-0.045	**-0.126**	**189**	366	457	0.346	0.346	0.334
	M	-9.722E-03	-1.721E-03	-0.010	**-0.017**	3568	2118	**1891**	0.037	0.026	**0.067**
	SD	0.016	**4.642E-03**	0.012	0.027	3509.766	1354.926	**982.427**	**0.065**	0.725	0.081
g04	B	-30655.331	-30665.439	-30665.539	-30665.539	0	0	0	0.110	0.120	**0.124**
	M	-30664.613	-30664.606	-30665.539	-30665.539	4	2	2	0.070	**0.080**	0.071
	SD	0.573	0.549	7.400E-012	7.400E-012	5.149	3.115	**2.921**	0.023	0.023	0.025
g05	B	-	-	5126.646 (18)	**5126.496**	33459 (12)	16845	**13087**	4.273E-07 (12)	0.087	0.087
	M	-	-	6057.259	**5140.060**	86809	23776	**17037**	1.250E-07	**0.056**	0.036
	SD	-	-	232.251	**15.525**	45359.759	3683.613	**2211.676**	**1.643E-07**	0.037	0.032
g06	B	-6959.517	-6959.926	-6958.704	**-6961.814**	**180**	256	254	0.799	**0.807**	0.772
	M	-6948.937	-6948.121	-6941.207	**-6961.814**	**440**	513	562	0.306	**0.348**	0.296
	SD	6.312	6.415	9.059	2.679E-04	273.858	258.324	**186.856**	0.207	**0.188**	0.193
g07	B	43.731	38.916	41.747	**24.444**	**178**	484	812	2.117	**2.504**	2.499
	M	68.394	64.186	59.077	**25.188**	**873**	1164	1316	1.656	1.919	**1.963**
	SD	40.698	17.156	7.653	**0.599**	711.998	401.181	**252.411**	0.363	0.369	**0.350**
g08	B	-0.095825	-0.095825	-0.095825	-0.095825	4	**0**	3	0.494	0.451	**0.556**
	M	-0.095824	-0.095825	-0.095825	-0.095825	**56**	78	76	0.317	0.304	**0.356**
	SD	1.751E-07	7.258E-08	4.234E-17	4.234E-17	**39.709**	52.159	57.963	0.091	0.092	**0.072**
g09	B	692.852	693.878	696.947	**680.637**	**5**	8	17	4.685	4.394	**4.768**
	M	713.650	708.274	728.730	**680.671**	**88**	94	107	**2.622**	2.209	2.510
	SD	12.969	10.155	15.804	**0.021**	50.144	**48.131**	69.187	1.298	**1.126**	1.246
g10	B	8024.273	8769.477	8947.646	**7097.001**	**242**	579	522	0.598	0.665	**0.678**
	M	8931.263	9243.752	9247.134	**7641.849**	**861**	972	1202	0.360	**0.482**	0.468
	SD	390.577	229.449	**184.691**	361.366	329.899	**269.375**	456.381	0.138	**0.107**	0.117
g11	B	0.749	0.749	0.750	**0.749**	**85**	249	364	0.143	0.143	0.143
	M	0.752	0.755	0.799	**0.749**	1662	1152	**1009**	0.088	**0.113**	0.101
	SD	9.273E-03	0.014	0.057	**1.999E-03**	2101.228	832.708	**572.870**	**0.047**	0.049	0.052
g12	B	-0.999	-0.999	-0.999	**-1.000**	2	**0**	1	**0.342**	0.281	0.285
	M	-0.999	-0.999	-0.999	**-1.000**	19	**15**	25	**0.136**	0.119	0.096
	SD	6.967E-07	5.137E-07	2.596E-05	**0.000**	**16.012**	17.784	21.712	0.072	**0.067**	0.074
g13	B	-	-	-	0.081	-	11497	**8402**	-	1.165	**2.327**
	M	-	-	-	0.454	-	17553	**12874**	-	0.410	**0.549**
	SD	-	-	-	0.258	-	2820.497	**1826.789**	-	0.346	**0.576**

Table 3. Statistical results of MPSO and PSO-based approaches. **Boldface** remarks the best result per function. The results with a "∗" seem to be infeasible, but they could not be verified in their original sources.

STATE-OF-THE-ART ALGORITHMS AND MPSO					
Problem & Bks.		Toscano & Coello [10]	Lu & Chen [13]	Cagnina et al. [12]	MPSO
g01	B	**-15.000**	-15.000	-15.000	**-15.000**
-15.000	M	**-15.000**	-14.418	-15.000	**-15.000**
	W	**-15.000**	-12.453	*-134.219	**-15.000**
g02	B	**-0.803432**	-0.664	-0.801	-0.802629
-0.803619	M	**-0.790406**	-0.413	0.765	-0.713879
	W	**-0.750393**	-0.259	0.091	-0.600415
g03	B	-1.004	-1.005	**-1.000**	-0.641
-1.000	M	-1.003	-1.002	**-1.000**	-0.154
	W	-1.002	-0.934	**-1.000**	-3.747E-03
g04	B	-30665.500	**-30665.539**	-30665.659	**-30665.539**
-30665.539	M	-30665.500	**-30665.539**	-30665.656	**-30665.539**
	W	-30665.500	**-30665.539**	-25555.626	**-30665.539**
g05	B	5126.640	**5126.484**	5126.497	5126.498
5126.498	M	5461.081	5241.054	5327.956	**5135.521**
	W	6104.750	5708.225	*2300.5443	**5169.191**
g06	B	-6961.810	-6961.813	-6961.825	**-6961.814**
-6961.814	M	-6961.810	-6961.813	-6859.075	**-6961.814**
	W	-6961.810	-6961.813	64827.544	**-6961.814**
g07	B	24.351	**24.306**	24.400	24.366
24.306	M	25.355	**24.317**	31.485	24.691
	W	27.316	**24.385**	4063.525	25.15
g08	B	**-0.095825**	**-0.095825**	-0.095825	**-0.095825**
-0.095825	M	**-0.095825**	**-0.095825**	-0.095800	**-0.095825**
	W	**-0.095825**	**-0.095825**	-0.000600	**-0.095825**
g09	B	680.638	**680.630**	680.636	680.638
680.630	M	680.852	**680.630**	682.397	680.674
	W	681.553	**680.630**	18484.759	680.782
g10	B	7057.900	**7049.248**	7052.852	7053.963
7049.248	M	7560.047	**7049.271**	8533.699	7306.466
	W	8104.310	**7049.596**	13123.465	7825.478
g11	B	0.749	**0.749**	0.749	0.749
0.749	M	0.750	**0.749**	0.750	0.753
	W	0.752	**0.749**	0.446	0.776
g12	B	**-1.000**	**-1.000**	-1.000	**-1.000**
-1.000	M	**-1.000**	**-1.000**	-1.000	**-1.000**
	W	**-1.000**	**-1.000**	9386	**-1.000**
g13	B	0.068	**0.053**	0.054	0.066
0.053949	M	1.716	0.681	0.967	**0.430**
	W	13.669	2.042	1.413	**0.948**

consistent and in presence of equality constraints, MPSO is more competitive. Furthermore, the dynamic parameter control added to MPSO was beneficial because the original LBPSO with constriction factor was unable to reach the feasible region faster in all test problems. Finally, the ability of PSO to move inside the feasible region was improved in MPSO. Another interesting aspect here is that local best PSOs (both MPSO and LBPSO) were more competitive to move inside the feasible region than GBPSO.

As a final experiment, a comparison against state-of-the-art PSO-based approaches is presented in Table 3. Best (B), Mean (M) and worst (W) results from a set of 30 independent runs by MPSO, using the same parameter values of the previous experiments are presented. The results are compared against: A) Toscano & Coello PSO [10], B) Lu & Chen PSO [13] and C) Cagnina et al. PSO [12]. The results used in this experiment were obtained from their original papers i.e. we conducted an indirect comparison.

The performance of MPSO is more consistent with respect to the results of the compared approaches in problems g05 and g13, both with nonlinear equality constraints. Problem g05 has a combination of equality and inequality constraints. Thus, it seems that the separation of sums of constraint violation provides more specific information to guide the search. However, this statement requires further studies. MPSO reached consistently the best known solution in five problems: g01, g04, g06, g08 and g12 and was very competitive in five more: g02, g07, g09, g10, g11. The test function where MPSO did not provide competitive results was problem g03 (ten decision variables and one nonlinear equality constraint). It seems that the combination of high dimensionality with equality constraints may lead to a premature convergence of the approach. The results of this last experiment show that MPSO provides a competitive performance against PSO-based state-of-the-art approaches. It is worth mentioning that MPSO does not use additional operators and/or parameters like the compared approaches and the number of evaluations required to reach these results was 160,000, compared to 340,000 required by Toscano & Coello [10] and Cagnina et al. [12], only above the 50,000 required by Lu & Chen [13] where an additional parameter is added to the flight formula and there is no clear evidence about its fine-tuning.

5 Conclusions and Future Work

In this paper, an analysis of the behavior of PSO in constrained search spaces was presented. Four original variants were compared in a well-known benchmark by using a simple parameterless constraint-handling technique. Based on statistical results, the most competitive variant was detected (Local best with constriction factor). A dynamic parameter control mechanism was added to update two PSO parameters (k and c_2) and the constraint-handling mechanism was improved in its third criterion by using separate sums of constraint violation for equality and inequality constraints. This modified PSO (MPSO) was compared with two original PSO variants using two performance measures as to analyze how fast they reach the feasible region and also their ability to move inside it. Finally, the MPSO was compared with three state-of-the-art PSO-based approaches for constrained optimization. The study of PSO original variants, as a first step of design, allowed to get a PSO-based approach to solve constrained optimization problems without adding extra parameters or other mechanisms which may eliminate PSO simplicity. The performance measures used in the second experiment showed that

MPSO is faster to reach the feasible region of the search space in presence of equality constraints and also that its ability to improve feasible solutions previously found was enhanced. Finally, the performance of MPSO was very competitive against algorithms representative of the state-of-the-art.

The future work consists on testing MPSO in more problems with a combination of equality and inequality constraints and to solve engineering design problems.

References

1. Eiben, A., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
2. Engelbrecht, A.: Fundamentals of Computational Swarm Intelligence. Wiley, Chichester (2006)
3. Dorigo, M., Maniezzo, V., Colorni, A.: The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man and Cybernetics-Part B 26, 29–41 (1996)
4. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6, 58–73 (2002)
5. Michalewicz, Z., Schoenauer, M.: Evolutionary Algorithms for Constrained Parameter Optimization Problems. Evolutionary Computation 4, 1–32 (1996)
6. Parsopoulos, K., Vrahatis, M.: Unified Particle Swarm Optimization for solving constrained engineering optimization problems. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 582–591. Springer, Heidelberg (2005)
7. Li, X., Tian, P., Kong, M.: Novel particle swarm optimization for constrained optimization problems. In: Zhang, S., Jarvis, R. (eds.) AI 2005. LNCS (LNAI), vol. 3809, pp. 1305–1310. Springer, Heidelberg (2005)
8. Krohling, R.A., dos Santos Coelho, L.: Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. IEEE Transactions on Systems, Man and Cybernetics Part B 36, 1407–1416 (2006)
9. He, S., Prempain, E., Wu, Q.H.: An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems. Engineering Optimization 36, 585–605 (2004)
10. Toscano-Pulido, G., Coello Coello, C.A.: A Constraint-Handling Mechanism for Particle Swarm Optimization. In: Proceedings of the Congress on Evolutionary Computation 2004, Piscataway, New Jersey, vol. 2, pp. 1396–1403. IEEE Service Center, Los Alamitos (2004)
11. Liang, J.J., Suganthan, P.N.: Dynamic Multi-Swarm Particle Swarm Optimizer with a Novel Constrain-Handling Mechanism. In: 2006 IEEE Congress on Evolutionary Computation (CEC 2006), Vancouver, BC, Canada, pp. 316–323. IEEE, Los Alamitos (2006)
12. Cagnina, L.C., Esquivel, S.C., Coello, C.A.C.: A Particle Swarm Optimizer for Constrained Numerical Optimization. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 910–919. Springer, Heidelberg (2006)
13. Lu, H., Chen, W.: Dynamic-objective particle swarm optimization for constrained optimization problems. Journal of Combinatorial Optimization 12, 409–419 (2006)
14. Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering 186, 311–338 (2000)
15. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley, Chichester (2001)
16. Mezura-Montes, E., Coello, C.A.C.: Identifying On-line Behavior and Some Sources of Difficulty in Two Competitive Approaches for Constrained Optimization. In: 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1477–1484. IEEE Press, Los Alamitos (2005)

An Approach to Searching for Two-Dimensional Cellular Automata for Recognition of Handwritten Digits

C.C. Oliveira Jr.[2] and P.P.B. de Oliveira[1,2]

Universidade Presbiteriana Mackenzie
[1] Faculdade de Computação e Informática
[2] Pós-Graduação em Engenharia Elétrica
Rua da Consolação 896 - Consolação
01302-907 São Paulo, SP – Brazil
carloscandidojr@gmail.com,
pedrob@mackenzie.br

Abstract. One of the contexts in which cellular automata have clearly demonstrated their effectiveness has been in problems involving strong and explicit spatial constraints, as happens in pattern formation and growth. By analogy, attempts to use cellular automata in pattern recognition have also been used in the literature and some progress has been made. However, in general, they still represent more of an unfulfilled promise, due to the lack of a recognition model which cellular automata would naturally fit in, the lack of effective ways to implement it, and the lack of generality of the available approaches. Here, experimental results are reported in the direction of using cellular automata in the task of handwritten digit recognition, in which an evolutionary algorithm searches for two-dimensional cellular automata rules that would transform a given digit image into a match, as close as possible, to a prototype image of that family, so that, the closer the match, the better the recognition of the input image. Although the results reported might still fall shorter than consolidated commercial techniques for the task, the approach presented is quite attractive in terms of the efficacy level it allowed to achieve, and because of its simplicity, which suggests a potential generality from the perspective of its use in other domains.

Keywords: Cellular automaton, pattern recognition, handwritten digit, handwritten character, evolutionary computation, genetic algorithm.

1 Introduction

Cellular automata (CAs) are distributed systems comprising a number of typically identical simple components, with local connectivity over a regular lattice, whose global configuration changes over time, according to a local state transition rule [17]. Quite remarkable about them is that they can be regarded both as an universal computing device, and as a dynamical system, discrete in space, time and state variables [17].

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 462–471, 2008.

One of the contexts in which cellular automata have clearly demonstrated their effectiveness has been in problems involving strong and explicit spatial constraints, as happens in the growth of urban settlements [14]; in the context of natural hazards in the environment due to the dispersion of, say, forest fire [25] or volcanic lava [27]; propagation of contagious diseases [26]; and pattern formation in biology [24] or engineering [21, 22]. And based upon their computing abilities, one might wonder, analogously, about using them also as pattern recognisers of spatial data.

Attempts to use cellular automata in pattern recognition in general, mainly spatial constrained data, have been a recurring theme in the literature, so that some progress has indeed been made [2, 3, 5, 8, 19, 20]. However, in general, they still fall short of the requirements for real-world, effective applications, thereby constituting more of an unfulfilled promise. These shortcomings have been due to the lack of a recognition model which cellular automata would naturally fit in, the lack of effective ways to implement it, and the lack of generality of the available approaches. Symptomatic of this situation is, for instance, the fact that, out of the 117 citations in [28], and out of the 255 citations in [29], none of them refer to cellular automata; also, out of the more than 88,000 references in [16], the joint search for ´cellular automata´ and ´cellular automaton´ entries return only 281 hits from the database, a mere 0.3%.

Recognition of handwritten digits by computational systems is a non-trivial task, due to the huge diversity of forms characters may display. In fact, the strong spatial constraints embedded in them are essential and inherent to their nature, so that they should thus be accounted for by any recognition system. Despite this difficulty, the task has been the subject of extensive studies, to the extent that it can be considered a well solved task presently, with even very effective commercial products available, based on consolidated techniques; nevertheless, contrarily to what one might think, new techniques continue to be proposed, with hopes for increased performance and generality [6, 12, 13, 18].

All this context renders the recognition of handwritten digits quite appealing and challenging for a cellular automaton based approach, which is precisely the motivation of the present work; here, an evolutionary algorithm searches for two-dimensional cellular automata rules that would transform a given input digit image into a match, as close as possible, to a prototype image of that family, so that, the closer the match, the better the recognition of the input image.

Although the achievements reported might still fall shorter than other existing competing models for the task, the computational technique presented is quite attractive not only in terms of the efficacy level it allowed to achieve, but also because of its simplicity, which suggests a potential generality from the perspective of its use in other domains.

In the next section the recognition approach at issue is introduced. Section 3 then describes the experiments performed and Section 4 the results obtained. The paper closes with Section 5, with an overall appreciation of the work, including a discussion on its roots and its natural follow-up.

2 The Approach

2.1 Cellular Automata as Dynamical Processors of Spatial Patterns

Cellular automata (CAs) are distributed systems comprising a large number of identical simple components with local connectivity [17]. These structures, the cells, form an n-dimensional grid, and can take on any of k possible states that transform over time according to a local rule or function. This rule determines the state of every cell at instant t+1 according to the states of its neighbouring cells at time t. The number of cells in the neighbourhood is usually determined by the radius r of the rule, which is the distance between the cell at issue and the farthest cell in its neighbourhood.

For given values of k and r, the number of possible rules defines the rule space, where such rules are usually enumerated according to Wolfram´s lexicographical order [17]; in the case of binary CAs, the latter is such that the output bits of each rule are arranged from neighbourhood 111…1 to 000…0. For instance, the rule space of the ´elementary´ CAs is formed by all 256 possible rules with k=2 and r=1, its rule 232 being represented by the binary number 11101000 formed out of its output bits, as illustrated in Figure 1, where whites and blacks represent 0s and 1s, respectively.

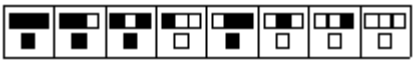


Fig. 1. Representation of rule 232 of the elementary space

A two-dimensional cellular automaton consists of a two-dimensional grid, with state transitions defined according to a two-dimensional neighbourhood. The most widely used neighbourhoods are Moore neighbourhood, that considers a cell and all its eight immediate neighbours, and von Neumann neighbourhood, which incorporates only a cell and its four immediate vertical and horizontal neighbours.

In successful approaches to the problem of handwritten digit recognition, computational systems typically allow input of a standard form and subsequently compare a new input with a standard prototype, explicitly or implicitly acquired, through learning, in a previous stage [1].

Likewise, here the handwritten digits are represented as a two-dimensional, binary cellular automaton initial configuration, with black pixels standing for 1 and white pixels for 0. A CA rule is then applied that leads the initial configurations representing digits, to corresponding final configurations, associated to the classes of digits, from 0 to 9.

All experiments reported below relied upon input digits comprising a set of well-known images of handwritten digits in the literature [10], as exemplified in Figure 2. Handwritten digit families were then defined, containing groups of handwritten digits representing the same digit, and the families organised in distinct sizes, according to the number of images within it. For example, a digit family for digit 3 and size 5 is a group formed by 5 images of digit 3, as depicted in Figure 3.

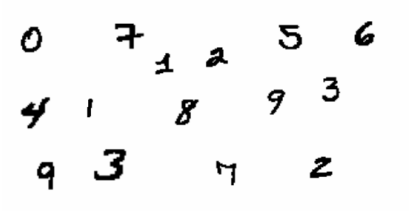


Fig. 2. A sample of initial configurations

Fig. 3. A digit family of size 5 for digit 3

2.2 The Recognition Model

A key notion in the recognition model developed is that of a *prototype*, which is defined as a particular image that might represent all the images in a digit family. In order to calculate the prototype image, it suffices to calculate the mean image within a group, so that each point in the prototype represents the arithmetic mean of the corresponding points across all images at issue.

As suggested earlier, recognition is deemed effective when, upon applying a given CA rule to initial configurations representing the same digit, the final configurations achieved by the CA lattice preserve some similarity with each other, according to the digit family the input belongs to. Such similarity is verified by calculating the distance between the final configuration generated from a given input digit, and the prototype of the final configurations that has been previously calculated for the handwritten digit family of the input digit. Naturally, ten prototypes needed to be calculated, one for each digit.

Empirical trials in the problem led to Euclidean distance being employed as the basis for the similarity distances between the images, mainly because of its widespread use.

Recognition of a specific handwritten digit is then granted if its distance to the prototype of the family (or class) it is supposed to belong to is the smallest. This means that the cellular automaton rule employed to process the given initial configuration (the input handwritten digit) has led to a final configuration that is close, in Euclidean distance, to the prototype of the family the input digit belongs to.

For the generation of the prototypes, an image set was established, the ´training´ set; analogously, a test set of images was also defined, consisting of those used to verify the effectiveness of the approach. The creation of the training and test sets relied upon 1500 images that were initially selected from [10], with the constraint of their not being terribly ambiguous to humans. Then, 10% of them were drawn to form the test set, 15 for each digit class. The remaining subset of 1350 images was further subdivided into 5 groups, the first one having 270 images and the subsequent ones obtained by adding 270 new images to the preceding one.

Various recognition trials were carried out with the previously defined image sets; however, considering that each new sub-group of the training set that was added to the previous only incremented recognition slightly, the smaller group of 270 images (27 from each digit class) ended up being employed for training in the experiments reported below, with the gain of a much smaller computation time.

3 Experiments and Results

3.1 Recognition with Rules Enumerated through Crossed ECAs

The two-dimensional CA space delimited by Moore neighbourhood was defined as the target of the searches to be carried out. This choice derived from the huge range of possibilities it supports, comprising a total of 2^{512} rules, that even includes the other natural candidate, von Neumann neighbourhood space.

However, before embarking on the evolutionary searches in the chosen target CA rule space, an initial enumerative experiment was carried out in a much smaller space, that can be shown to constitute a subspace of Moore neighbourhood, the one that is brought about by what is named here the ´crossed elementary cellular automata´ (Crossed ECAs).

Crossed ECAs entail application of two-dimensional cellular automata rules, through the use of the application of elementary CA rules, in two stages, as follows:

1. An elementary rule is applied to the rows of the two-dimensional lattice;
2. Another elementary rule (possibly the same one as before) is applied to the columns of the lattice, using the outcome of the previous stage as input.

In the application of ECA rules to rows and columns at the two stages, the two-dimensional cellular automaton is partitioned and considered a one-dimensional elementary cellular automaton with periodic boundary conditions. Each application (rows and columns) is then considered a separate CA iteration.

The advantage of such a scheme is that it samples a meaningful subset of all possible two-dimensional CAs of interest, in that it allows transfer of knowledge of elementary CAs to Moore neighbourhood CAs. To enable this, a set of all ordered ECA rule pairs was formed consisting of the pairwise grouping of elementary rules, excluding those classified as complex (i.e., edge-of-chaos) or chaotic, according to Li-Packard classification [11]. These exclusions, associated to the individual rules´ dynamical behaviours aimed at preventing the effect of an excessive convergence time in the temporal evolutions, or even the lack of convergence at all.

The ordered pairs were obtained by enumeratively combining the individual elementary CA rules, one by one, without repeating rules, resulting in 5329 rule pairs; each one was then applied to a handwritten digit following the Crossed ECA structure.

The 5329 rule pairs were then submitted to the recognition process, according to the following sequence: generation of the 10 prototypes, by applying the rule pair to the training set; generation of the final configuration of each image in the test set, by applying the rule pair for a certain number of time steps; calculation of the Euclidian distances between each final configuration and the 10 prototypes; and, verification of the minimum distances, so that a given input configuration was granted recognised if its minimum distance corresponded exactly to the expected prototype for it.

After running the process for all 5329 rule pairs, the optimal rule pair 62-168 recognised 95.3% of the entire test set, much higher than the average of the set. This corresponded to recognition of 143 images from the test set, the pair having reached such a recognition rate by evolving the initial configuration for 2 iterations. Figure 4 shows the prototypes generated by that rule pair. It is interesting to realise that the

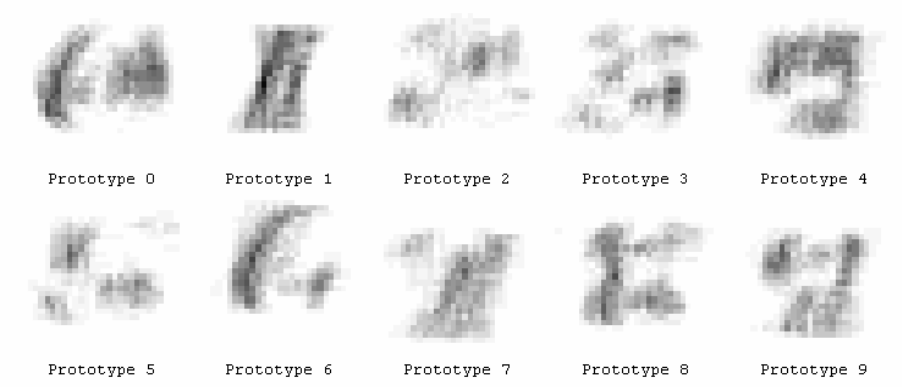


Fig. 4. Prototypes generated by the rule pair 62-168, each one corresponding to a digit family

generated prototypes have very little relation to the overall shape of each digit, even considering only 2 time steps; however, looking at them in more detail it seems that they indeed represent, even though vaguely, some key portion of each digit, as can be noticed, for instance, comparing the prototypes for digits 6 and 9, or those for digits 1 and 7. This strongly suggests that the rules seem to be doing a preprocessing of the input digit, so that they would become more amenable for recognition. But what is remarkable is that such a feature was not embedded in the system at all, explicitly or otherwise.

3.2 Recognition with Rules Searched by a Genetic Algorithm

Since the entire Crossed ECA space is rather small, composed of only $256^2=2^{16}$ rules, and the biased, enumerative experiment above showed that even in this space good rules do exist, better rules are likely to exist within the much larger Moore neighbourhood rule space.

In order to search for rules in this vast space, a genetic algorithm (GA) was developed, based on the one described in [9]. In the algorithm, each individual is the binary, 512 bit-long representation of one possible rule in Moore neighbourhood space. The representation followed Wolfram´s lexicographical ordering mentioned above, now applied in two dimensions.

The fitness evaluation of each candidate rule relies upon the same process described in the previous experiment, which considers the images in the training set. The actual value of the fitness of a candidate rule is then the recognition rate of the rule, over the images in the test set.

GA runs were devised so as to try a number of different parameters. Among all the runs, the optimal rule recognised 98% of the images in the test set, thus representing an improvement on the best rule pair of the enumerative experiment. This result was obtained using the following parameters: population size of 50 individuals (rules); single-point crossover, at 100% rate; mutation rate of 0.02% per bit; selection of 20% of the elite, the remaining 80% of the rules being randomly chosen from the entire population; CAs applied for 2 iterations; and 120 generations per run.

Recognition of 98% of the test set implied that only three digits were unrecognised, the ones depicted in Figure 5. Inspection of the unrecognised digits did not suggest any overall reason related to their recognition failure; however, it should be noticed that the unrecognised digit 2 is dubious even for human recognisers. Nevertheless, in view of the vast rule space concerned, a rule may exist that might allow a further increase in the recognition rate of the test set, maybe even leading to full recognition.

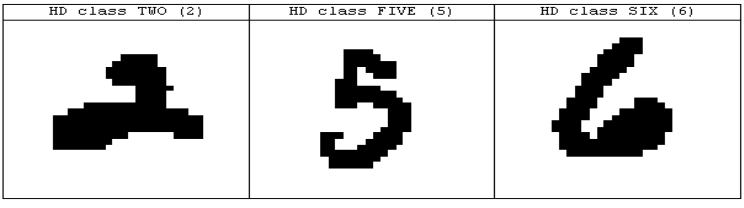


Fig. 5. Handwritten digits not recognised by the best rule found by the GA, and their respective digit family

For the purposes of evaluating the effectiveness of the results obtained, two baseline or reference experiments were carried out. The first one derived from the idea of investigating CA-based recognition from a spectral context, that is, to check the effect of handling images in the frequency domain, chiefly due to their non-susceptibility to translation and other transformations. To this end, all images underwent discrete Fourier transform, and, in order to quantitatively ascertain similarity between the images, Euclidian distance was used between the magnitude of their components, referred to as ´spectral similarity´. Although a range of evaluation approaches for the rules were tried out, the best result reached only 82% recognition efficacy.

In the second baseline experiment, CAs were not employed at all to transform the initial configurations. Accordingly, the prototypes were generated directly from the images of the training set, and the test set was then checked by calculating the Euclidian distance of each image and its respective prototype. Hence, the key difference of this experiment was that the handwritten digit images did not undergo the transformations entailed by the CA rule application. As a result, 88.7% of the test set was recognised, a surprisingly high value that suggests the effectiveness of the notion of prototype, as used here.

Altogether, the baselines demonstrate the improved performance entailed by the CA rule image transformation, linked to the notion of (non-spectral) prototype. Although such comparisons might lag behind of other recognition techniques, particularly those in commercial systems, they certainly hint at the strengths of the proposed approach.

4 Concluding Remarks

Various CA based efforts in image processing, particularly those in edge detection, are available in the literature [4, 15, 30, 31, 32] and, naturally, CA based recognition approaches based on edge detection as well [2, 33, 34]. However, while this type of

processing is explicitly targeted in the latter endeavours, here the CA rule being searched for has no ´obligation´ whatsoever of carrying out edge detection at any stage of its operation on the initial configuration. In fact, quite peculiar of the approach explored here is the total absence of a priori assumptions regarding the prototype nature, which does not have to abide by any predefined constraint. This departs this work the most from other related efforts. This might sound naïve at first glance, but this simple idea reflects the very essence of cellular automata nature, and, as such, the most natural way, we believe, towards their usage.

Such simplicity relies upon the notion of prototype (drawn from a rule's capacity to yield similar temporal evolutions to similar initial configurations of the same digit family), the reliance on a direct (non-spectral) approach for its creation, and use of plain Euclidean distance to drive the evaluation of a candidate rule. Although the present results may not have yet been entirely satisfactory, the conceptual structure stemmed from the work stands out in itself, both in effectiveness and simplicity, and paving the way for further enhancements.

The enumerative experiment relying on the elementary rule combinations showed to be relevant, in that the dynamical nature of the individual rules involved pointed at fruitful points in Moore neighbourhood space, as reflected by the good recognition efficacy achieved by rule pair 62-168. This realisation encouraged the search for rules in the complete space delimited by the Moore neighbourhood.

One possibility for further improvements in the efficacy level in the digit recognition task might be the idea of favouring the search towards rules that would be able to recognise simpler structures first, such as digits formed by straight lines only (namely, 1, 4 and 7) versus those fully rounded (i.e., digits 0, 3, 6, 8 and 9), or simple straight line segments versus curved segments. This might help the search to first tackle ´simpler´ problems, for only then to go freer; this is analogous to what has recently been done in [23], which was key to the results reported therein for the density and the parity problems, the best ones presently available for these tasks.

Further possible efficacy improvements might come from using much more sophisticated evolutionary algorithms than the very simple one described here; once again, [23] provides several good possible extensions. Quite encouraging in this respect is that, even with a simple evolutionary technique, good rules could be found.

A key issue in further developments of this work is to probe the scalability of the method, something that has not yet been looked at. Similarly, care should also be taken for the analysis of the unrecognised digits, since, as shown above, one of the test instances we used can be misleading even for human recognisers. Also, a proper, systematic comparison of the approach with existing consolidated techniques, even commercial ones, is tempting in the horizon, provided the required resources are made available.

Further topics suggestive of a distinct stance of investigation would include the use of non-uniform CAs, as in [19], in which distinct cells would be allowed to embed different rules, thus giving more flexibility for evolution to work, and the development of new applications, such as in recognition of other types of characters (letters or ideograms), thus allowing to probe the generality of the method, a feature that is certainly facilitated by its conceptual simplicity.

These are all appealing targets to be pursued in parallel and in sequence, whose ultimate goal is to put under scrutiny the usefulness of cellular automata as pattern

recognisers, possibly allowing them to re-enact, in the new role, their quintessential, well established vocation as pattern generators.

Acknowledgement

We are very grateful to MackPesquisa – Fundo Mackenzie de Pesquisa, for various grants, in particular, the most recent one, from Edital 2007; Wolfram Research, for a Mathematica Academic Grant (No 1149); and FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo, for the research grant Proc. 2005/04696-3.

References

1. Bahlmann, C., Burkhardt, H.: The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping. IEEE Transactions on Pattern Analysis and Machine Intelligence 6(3), 299–310 (2004)
2. Cagnoni, S., Mordonini, M.: Cellular automata and computer vision: Some case studies. AI*IA Notizie XIV(2), 23–31 (2001)
3. Raghavan, R.: Cellular automata in pattern recognition. Information Sciences 70(1-2), 145–177 (1993)
4. Popovici, A., Popovici, D.: Cellular automata in image processing. In: Proc. of the 15th Int. Symp. on the Mathematical Theory of Networks and Systems, Univ. of Notre Dame (2002)
5. Ganguly, N., Das, A., Maji, P., Sikdar, B.K., Pal Chaudhuri, P.: Theory and application of cellular automata for pattern classification. Fundamenta Informaticae 58(3-4), 321–354 (2003)
6. Kherallah, M., Haddad, L., Alimi, A.M., Mitiche, A.: On-line handwritten digit recognition based on trajectory and velocity modeling. Pattern Recognition Letters 29(5), 580–594 (2008)
7. Guan, S., Zhang, S.: An evolutionary approach to the design of controllable cellular automata structure for random number generation. IEEE Transactions on Evolutionary Computation 7(1), 23–26 (2003)
8. Hogg, T., Huberman, B.: Parallel computing structures capable of flexible associations and recognition of fuzzy inputs. Journal of Statistical Physics, 41–115 (1985)
9. Jacob, C.: Illustrating Evolutionary Computation with Mathematica. Morgan Kaufmann, San Francisco (2001)
10. Lecun, Y., Corte, C.: The MNIST database of handwritten digits (last access, April 2008), `http://yann.lecun.com/exdb/mnist`
11. Li, W., Packard, N.: The structure of elementary cellular automata rule space. Complex Systems (4), 281–297 (1990)
12. Zhang, P., Bui, T.D., Suen, C.Y.: A novel cascade ensemble classifier system with a high recognition performance on handwritten digits. Pattern Recognition 40(12), 3415–3429 (2007)
13. Lauer, F., Suen, C.Y., Bloch, G.: A trainable feature extractor for handwritten digit recognition. Pattern Recognition 40(6), 1816–1824 (2007)
14. Pinto, N.N., Antunes, A.P.: Cellular automata and urban studies: A literature survey. ACE: Architecture, City and Environment 1(3), 368–399 (2007)
15. Rosin, P.L.: Training cellular automata for image processing. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 195–204. Springer, Heidelberg (2005)

16. Price, K.: The annotated computer vision bibliography (last access, April 2008), http://www.visionbib.com
17. Wolfram, S.: A New Kind of Science, 1st edn. Wolfram Media (2002)
18. Savas, B., Eldén, L.: Handwritten digit classification using higher order singular value decomposition. Pattern Recognition 40(3), 993–1003 (2007)
19. Maji, P., Chaudhuri, P.P.: Non-uniform cellular automata based associative memory: Evolutionary design and basins of attraction. Information sciences 178(10), 2315–2336 (2008)
20. Maji, P., Chaudhuri, P.P.: Fault diagnosis of electronic circuits using cellular automata based pattern classifier. In: Prasad, B. (ed.) Soft Computing Applications in Industry, pp. 225–246. Springer, Berlin (2008)
21. Pikula, T., Gramatová, E., Fischerová, M.: Automatic design of cellular automata for generating deterministic test patterns. In: Digest of Papers: IEEE European Test Workshop (ETW), Maastricht, The Netherlands, pp. 285–286 (2003)
22. Dasgupta, P., Chattopadhyay, S., Chaudhuri, P.P., Sengupta, I.: Cellular automata-based recursive pseudoexhaustive test pattern generator. IEEE Transactions on Computers 50(2), 177–185 (2001)
23. Wolz, D., Oliveira, P.P.B.: Very effective evolutionary techniques for searching cellular automata rule spaces. Journal of Cellular Automata (forthcoming, 2008)
24. Deutsch, A., Dormann, S.: Cellular automaton modeling of biological pattern formation: Characterization, applications, and analysis. Birkhäuser, Boston (2005)
25. Encinas, A.H., Encinas, L.H., White, S.H., Rey, A.M., Sánchez, G.R.: Simulation of forest fire fronts using cellular automata. Advances in Engineering Software 38(6), 372–378 (2007)
26. White, S.H., Rey, A.M., Sánchez, G.R.: Modeling epidemics using cellular automata. Applied Mathematics and Computation 186(1), 193–202 (2007)
27. D'Ambrosio, D., Spataro, W.: Parallel evolutionary modelling of geological processes. Parallel Computing 33(3), 186–212 (2007)
28. Koerich, A.L., Sabourin, R., Suen, C.Y.: Large vocabulary off-line handwriting recognition: A survey. Pattern Analysis & Applications 6(2), 97–121 (2003)
29. Plamondon, R., Srihari, S.N.: Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 63–84 (2000)
30. Batouche, M., Meshoul, S., Abbassene, A.: On solving edge detection by emergence. In: Ali, M., Dapoigny, R. (eds.) IEA/AIE 2006. LNCS (LNAI), vol. 4031, pp. 800–808. Springer, Heidelberg (2006)
31. Goi, H.T., Perkowski, M.: An original method of edge detection based on cellular automata (last access, 2008), http://web.cecs.pdx.edu/~mperkows/temp/June16/A-New-Method-of-Edge-Detection-Based-on-Cellular-Automaton-Algorithm.doc
32. Slatnia, S., Batouche, M., Melkemi, K.E.: Evolutionary cellular automata based-approach for edge detection. In: Masulli, F., Mitra, S., Pasi, G. (eds.) WILF 2007. LNCS (LNAI), vol. 4578, pp. 404–411. Springer, Heidelberg (2007)
33. Namvong, A., Wongthanavasu, S.: Facial features localization from range images using cellular automata model. In: Proc. of IAIT 2007: Int. Conf. on Advances on Information Technology 2007, Bangkok, Thailand (2007)
34. Wongthanavasu, S., Tangvoraphonkchai, V.: Cellular automata-based algorithm and its application in medical image processing. In: Proc. of ICIP 2007: IEEE International Conference on Image Processing, vol. 3, pp. III.41–III.44 (2007)

Prediction of Pediatric Risk Using a Hybrid Model Based on Soft Computing Techniques

Yanet Rodríguez[1], Mabel González[1], Adonis Aguirre[2], Mayelis Espinosa[1], Ricardo Grau[1], Joaquín O. García[2], Luis E. Rovira[2], and Maria M. García[1]

[1] Centro de Estudios de Informática, Universidad Central de Las Villas, Cuba
[2] Hospital Pediátrico Universitario "José Luis Miranda" de Santa Clara, Villa Clara, Cuba

Abstract. We present an automatic system for the prediction of mortality risk in pediatric patients, which uses Soft Computing techniques instead of traditional ones based on score. The hybrid model applied combines both Case-Based Reasoning and Artificial Neural Networks with fuzzy set theory, taking its applications the advantages of these approaches. While the new way of prediction, named SAPRIM (Automated Predictor System of Infant Mortality Risk), was automatically defined from domain examples reducing the knowledge engineering effort, the experimental results using cross validation showed good accuracy with respect to other traditional classifiers. Besides, SAPRIM allows a more natural framework to include expert knowledge by using linguistic terms. After this automatic system was exploited by human experts for a year, the field evaluation corroborates good results.

1 Introduction

Diagnosis is the task of identifying the causes of a device malfunction and it can be formulated as: given a set of symptoms and a device's description, find an explanation to the symptoms. When the number of possible explanations is small the diagnosis can be tried as a classification problem, otherwise it can be seen as a problem of finding an explanation.

Different approaches to solve diagnosis problems exist, in which different types of knowledge and inference mechanisms are used. In the associative approach a knowledge that links the symptoms to defects in a direct way is used. In the approach based on models, a model of the equipment under consideration from which expectations can be derived is exploited. In the Rule-Based Systems, rules relate the symptoms and the explanations in a causal manner.

Human thought is not always based on logical reasoning. Sometimes it is needed to manage exceptions, in such applications Case Based Reasoning (CBR) [1] is recommended. That happens with many specialists in the area of diagnosis.

CBR solves non-structured problems using algorithms to determine a similarity measure between two objects [2]. A new problem is carried out in the context of known solutions for a set of previously solved problems of the application domain. The definition of the distance function plays an essential role to guarantee accurate solutions [3].

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 472–481, 2008.

Hybrid models, which combine different technologies to obtain a product that shares their advantages and minimizes their deficiencies, have received special attention. Measurements are crisp whereas perceptions are fuzzy. If a computational modeling uses words, then propositions drawn from a natural language, e.g., *small*, *high*, *old*, and the brain's crucial ability to manipulate perceptions will be used [4]. Many recent works corroborate the usefulness of combining both CBR and fuzzy logic to manage the uncertainty originated in CBR systems have been proposed [5] [6] [7], increasing the quality of a CBR system's outcome in the handling of real life ambiguous situations.

Several medical literatures refer the use Soft Computing paradigms [8] to medical diagnosis [9] [10]. In contrast, both PSI (Physiologic Stability Index) and PRIMS (Pediatric Risk of Mortality Score) [11] methods take the values of many variables, which are obtained from the patient's clinical and complementary tests, to evaluate the patient based on the score calculated from this data. This paper presents an automatic system for the prediction of mortality risk in pediatric patients, which is based on Soft Computing techniques instead of traditional procedures previously used, showing the advantages for the former.

2 Overview of the Hybrid Models Selected

A Hybrid model using soft computing techniques is tested for prediction of pediatric risk. It combines case-based reasoning (CBR) and artificial neural network (ANN) with fuzzy sets for the development of knowledge-based systems. Numeric attributes are modeling in terms of fuzzy sets. Thus, by using Fuzzy Set theory [12] the use of natural language mapping numeric data into linguistic terms is allowed. For example, the figure 1 shows a fuzzy partition defined for the attribute "temperature of a person". It is usually for physicians referred to "fever of a patient" by using words instead of numbers. Thus, a linguistic term is usually used (i.e. *low*, *medium*, *high*).

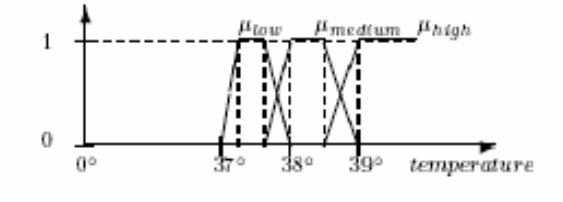


Fig. 1. Fuzzy representation for the attribute "Fever"

The problem solver can be either the Fuzzy-ANN model (referred as Fuzzy-SIAC) [13] or the CBR if the Connectionist Fuzzy Case-Based Reasoning Model (referred as ConFuCiuS) [14] is used. As no learning algorithm will performance best for all applications, the hybrid model's authors showed in [15] using UCIMLR data sets that either the CBR or the connectionist approach as problem solver can provide good results depending on memory of instance given. Even more, at least one variant provide results as well as C4.5. Additionally, the developed expert systems by using both variants of the hybridization take the advantages of techniques used.

2.1 The Fuzzy Associative ANN

The Fuzzy-SIAC is a fuzzy implementation of the Simple Interactive Activation and Competition model proposed in [16], capturing the merits of fuzzy set theory and this model of ANN. There exists a group of neurons for each attribute a, where a single neuron is allocated for each representative value A defined for an attribute a. If the attribute is numeric, representative values are linguistic terms, otherwise they are linguistic labels.

Arcs are established between all neurons in different groups. A non-supervised learning is applied to achieve the arc's weight $w_{A,B}$ between the representative values A and B is used. A measure based on fuzzy relative frequency is used. The Fuzzy-SIAC completes the pattern corresponding with the query q taking into account the degree of activation of the neurons in the group of the target attribute t. The representative value T_j corresponding with the node of greatest activation will be the value given to this attribute.

CBR can be classified into two major types: problem solving CBR and interpretative CBR [2]. After the fuzzy ANN model solved the problem, the last CBR functionality to avoid the non-existing explanation facilities of the connectionist approach is exploited. Thus, the fuzzy CBR component allows justifying the solution given by ANN with old experiences stored in the case base.

2.2 Connectionist Fuzzy CBR Model

Approaches based on similarity as CBR requires the definition of the similarity (or distance) function. After the previously explained Fuzzy-ANN model is trained, its weights and the membership degrees in the training examples are used to automatically generate a local distance function and an attribute weighting scheme. Besides, when two examples x and y are compared, a measure of "how close" these two values are can be obtained considering the difference between their membership degrees to each defined linguistic terms. Thus, the retrieval module of ConFuCiuS follows the Nearest Neighbor rule using a dissimilarity function automatically built from the memory of instances given. In other words, ConFuCiuS uses a local comparison criterion defined from knowledge encoded in the Fuzzy-SIAC model instead of HEOM (Heterogeneous Euclidean-Overlap Metric) used by the standard k-NN.

3 Prediction of Pediatric Risk Using the Hybrid Fuzzy-CBR-ANN Model

3.1 Problem Description

Elevating the quality of medical care constitutes one of the specific directives of the strategy that is reiterated in the policies of the ministry of Public Health of our country; within which pediatric attention has great importance and priority. A doctor's action before his patient includes, besides arriving to the diagnosis, emitting a prognosis on the possible evolution of the illness and its level of severity. Attending to this, the destination ward is oriented if the patient is tributary of entrance. There exist three scales of attention: conventional, intermediate care and intensive care; and the diagnostic and/or therapeutic procedures are different in each service.

In particular, one is required to decide in an appropriate and quick way if the patient should be admitted in the Intensive Care Unit (ICU), where those children who have a delicate state of health and a greater risk of dying are attended to. The specialists of this service are not always satisfied with the proposed both foreign models PSI and PRIMS. The PSI includes laboratory exams that are very expensive to carry out and traumatic for the patient, that would also slow the definition of an initial behavior to follow. The application of PRISM, although it is a simplification of the PSI that uses only 14 variables, looks at a minimum of five laboratory exams [17].

PRISM is a tool that is only valid in Europe and North America and therefore one cannot guarantee satisfactory results when it is used in other latitudes. A study carried out in a Pediatric ICU[1] pitched that when evaluating PRISM in a group of patients that died later on, only 17.5% were given as high risk of dying. The remaining patients were placed in the scale of low and moderate risk of mortality. We can conclude that PRISM did not perform well predicting the severity in this Pediatric ICU, which is a serious deficiency to take into account.

For that reason it was necessary to create a suitable prognosis model that helps in decision making in hospital administration to assist the specialist in determining the necessity or none necessity of the admittance of a patient in the intensive care ward. For this only the clinical and epidemic data of the patient are assessed, while laboratory exams to supplement the assessment of the current state of this patient are carried out.

3.2 Computational Solution Using Machine Learning

The computational model of the problem required theoretical knowledge acquired by specialists, and essentially the experience accumulated with practice. Nevertheless, the knowledge acquisition is the bottleneck problem to develop an expert system. As data of clinical histories of patients was taken, we applied techniques for learning models from sample data.

A group of experts has worked this hypothesis from a medical point of view (to infer the gravity of the patient's situation using clinical and epidemiological variables). These medical investigations indicated that 33 patient's characteristics (attributes) should be considered to decide whether or not the presented case requires admittance in the intensive care ward. A summary is given in Table 1.

Table 1. Summary of attributes

Type	Amount	Example	Domain
Nominal	23	Bleeding	Yes, No
Numerical	3	age (months)	0-192
String	1	#Clinical History	6 digits
Set	6	Skin color	jaundiced, pale, ...

Using the gathered sample, and with the help of the specialists, a case base of 1079 examples (538 "under observation" and 541 "serious-critical") was defined. Each case is characterized by the 33 attributes defined, one of which refers to the objective

[1] Hospital Pediátrico Universitario "José Luis Miranda" de Santa Clara, Villa Clara, Cuba.

attribute (class) defining as its possible values: "Under observation" and "Serious-critical"; where the first classification means that the patients should not be entered in an ICU, and in the remaining cases they should. As we expected, here the absence of some data is not uncommon.

From computational point of view, the patients' clinical histories are considered to define the examples that the system will use in its learning process (training set o case base). As a notation we use $\boldsymbol{M}$= *{(*$\boldsymbol{d}_1$, y_1*), (*$\boldsymbol{d}_2$, y_2*),.., (*$\boldsymbol{d}_n$, y_n*)}* to denote it, where for *i-th* instance (patient), $i = 1,..,n$, $\boldsymbol{d}_i \in \Re^m$ is referred to the description of the instance (patient's description) and y_i is their corresponding class output (patient's diagnosis). This application is a classification task since for each training case we know de class value according to the experts' assessment, and we need to discriminate new cases. Let $\boldsymbol{d}$= *(*d_1, d_2,..,d_m*)* be the description of an instance and a query q=(q_1,q_2,…,q_m), where $d_j \in X_j$ is the *j*-th attribute domain , j= 1,…,m.

The probability P(C/q) of assigning class C to a vector q depends on the procedures used in the construction of the model and its adaptive parameters or degrees of freedom [18]. Using this case base (the memory $\boldsymbol{M}$), the selection of the problem solver to infer the class is carried out. To do this, we make use of the Waikato Environment for Knowledge Analysis (WEKA[2]), a machine leaning tool where the classic classification algorithms and the variants of the hybrid model afore mentioned are implemented; as well as facilities for a comparative evaluation between several models.

Firstly, Table 2 shows the results achieved by the two variants of the hybrid model. Several measures of the classifier performance were considered depending on the experiment goal: the percentage of cases correctly classified (accuracy), the misclassification rate (square mean error-SME), Kappa statistic and other ones specified in the first column and recommended in such tasks. Each cell (for a model and a measure) is referred to average value for this measure computed after the construction of the model was repeated ten times for the instance memory $\boldsymbol{M}$ using 10-fold cross validation.

Table 2. Comparison of the two variants of the hybrid model

MEASURE\MODEL	FUZZY-SIAC	CONFUCIUS (κ=3)
Accuracy	84.37	**90.53**
Kappa	0.69	**0.81**
SME	0.49	**0.27**
SPECIFICITY	0.79	**0.86**
VPP	0.81	**0.87**
SENSITIVITY	0.90	**0.95**
F-MEASURE	0.85	**0.91**
ROC	**0.95**	**0.95**
VPN	0.89	**0.95**

As you can note the best results were achieved by ConFuCiuS (k=3) which refers to the connectionist fuzzy model to develop Case Based Systems, which uses CBR as problem solver. In other words, results are much better in favor of the ConFuCiuS

[2] WEKA is a Java-written open source. It is available at http://www.cs.waikato.ac.nz/~ml/weka/ under the GNU General Public License.

since it shows the lowest misclassification rate and the highest values for the rest of measures considered. As it is an instance of the *k*-NN, *k*=3 refers to the amount of similar cases (neighbors) to consider in the prediction function. Previously, numeric attributes were modeling as linguistic variables defining a fuzzy partition [19].

Table 3 shows the results from the comparisons carried out among ConFuCiuS and some classifiers implemented in the machine learning tool.

Table 3. Comparison of the results using several classifiers

MODEL	CONFUCIUS (*K*=3)	RL	C4.5	*K*-NN (*K*=3)
Accuracy	**90.53**	87.26	89.69	87.93
Kappa	**0.81**	0.75	0.79	0.76
SME	**0.27**	0.35	0.30	0.31
SPECIFICITY	0.86	0.86	**0.90**	0.77
VPP	0.87	0.86	**0.90**	0.81
SENSITIVITY	**0.95**	0.89	0.89	0.98
F-MEASURE	**0.91**	0.87	0.90	0.89
ROC	**0.95**	0.92	0.91	0.93
VPN	**0.95**	0.89	0.90	0.98

In the Table above, the third column is referred to Logistic regression (LR) [20], a multinomial logistic regression model with a ridge estimator (Logistic in Weka); the fourth refers to C4.5 [21], an inductive decision tree algorithm (J48 in Weka package); while the last contains the result achieved with the standard *k*-NN (IBL en Weka).

It is appreciated that ConFuCiuS in comparison with other well-known classifiers indicates the feasibility of its application to solve the application before described (prediction of pediatric risk). Although both ConFuCiuS and *k*-NN are instances of the *k*-NN, their differences are because of the similarity criterion influences decisively in which ConFuCiuS selects an exit generalized on to another one, and therefore in its performance.

3.3 Feasibility of the Prediction Using SAPRIM

The automatic system developed from ConFuCiuS model was named SAPRIM (acronym for Spanish translation of *Automated Predictor System of Infant Mortality Risk*), which uses Soft Computing techniques reviewed in the previous section. The programming language used was Java since the methods used in Weka were already implemented in this language, facilitating the development of the project.

The use of ConFuCiuS not only allows us obtain the best performance as we showed in the section above, we take some advantages of the hybridization used based on soft computing techniques. The comparison criterion used is to recover similar cases was automatically defined from examples. Thus, the use of ConFuCiuS allows developing a smart expert system, reducing the knowledge engineering effort. Knowledge engineering has been recast as case engineering.

On the other hand, numeric predictive attributes in terms of fuzzy sets were modeled. As such, representative values for numeric attributes are linguistic terms, facilitating the use of natural language more closely to human experts than numbers, and accounting for words with ambiguous meanings.

When the characteristic for a new pediatric patient are input to SAPRIM, it firstly allows us to determine if his state is either "Under observation" or "Serious-critical". Even more, it gives the option of to visualize the most similar cases, storied in the case base, with respect to the case problem. See Figure 1. In other words, if the user requires it, SAPRIM also shows the likeness of the most similar patients seen in the past with respect to the patient just evaluated by the software. Thus, human experts can understand the solution given by SAPRIM. This is in fact the "justification", the other advantage of to use ConFuCiuS model, mainly the Fuzzy-CBR approach.

Additionally, SAPRIM has the possibility of visualizing the influence (importance) of each predictive attribute value on the prognosis obtained; taking now advantages of the Fuzzy-ANN approach. In other words, the connectionist approach allows using the ANN' weight (knowledge encoded) of the arc between the processing neurons referred to predictive and target attribute values respectively. Note that for a predictive numeric attribute, the membership degrees of the value given in the description to the linguistic terms defined are considered. Thus, the characteristics known for the patient under evaluation by SAPRIM are presented to the user according its relevance in the context of last experiences with the same diagnosis.

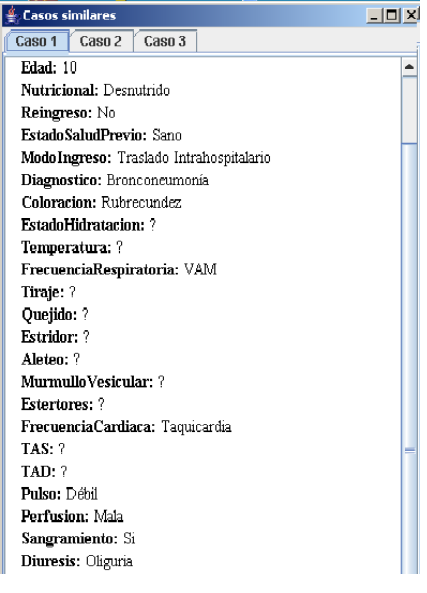


Fig. 1. The SAPRIM windows to visualize the most similar cases

3.4 Field Evaluation

After SAPRIM was exploited by human experts for a year, the field evaluation corroborates good results. We consider 99 cases evaluated by SAPRIM, and afterwards these same cases were blind evaluated by several specialists. We want to demonstrate the hypothesis that "SAPRIM predicts mortality risk as well as human expert do".

Seven pediatricians were interviewed, each one (named Sp.#) defining his individual prognosis for each patient ("under observation", "serious-critical"). To evaluate the results obtained by the system in time of exploitation in comparison with the specialists, a statistical analysis was made from the results given by the system. The Cochran test was applied to these answers showing that two groups of specialist can be formed according to their criteria. In other words, there was similarity between the answers of specialists {Sp.2, Sp.1, Sp.6} and between specialists {Sp.6, Sp.5,

Table 4. Statistics of the answers of specialists Sp.2, Sp.1 and Sp.6 (first group)

A. Frequency values

	"UNDER OBSERVATION"	"SERIOUS-CRITICAL"
SP.1	15	84
SP.2	9	90
SP.6	18	81

B. Results of the Cochran test

N	99
COCHRAN'S Q	5.250
DF	2
ASYMP. SIG.	.072

Table 5. Statistics of the answers of specialists Sp.6, Sp.5, Sp.7, Sp.3 and Sp.4 (second group)

A. Frequency values

	"UNDER OBSERVATION"	"SERIOUS-CRITICAL"
SP.6	18	81
SP.3	27	72
SP.4	28	71
SP.5	26	73
SP.2	9	90
SP.7	26	73

B. Results of the Cochran test

N	99
COCHRAN'S Q	6.809
DF	4
ASYMP. SIG.	.146

Table 6. Statistics of the answers of SAPRIM and specialists of the first group

A. Frequency values

	"UNDER OBSERVATION"	"SERIOUS-CRITICAL"
SP.6	18	81
SAPRIM	17	82
SP.1	15	84
SP.2	9	90

B. Results of the Cochran test

N	99
COCHRAN'S Q	6.031
DF	3
ASYMP. SIG.	.110

Sp.7, Sp.3, Sp.4}. The results are shown in the Tables 4 and 5. As you can note, Table 4 is referred to the first group of specialists while Table 5 resume the same results for the second group.

When the results given by the SAPRIM were included, the Cochran test showed concordance only with the first group (Table 6), which was verified to be formed by specialists of great experience including the expert that directs medical investigation.

Then the McNemar test was applied to compare the results of specialists 1, 2 and 6 with those of SAPRIM. Table 7 shows the final contingency table for the 99 cases, considering as desirable the results obtained from the consensus of the specialists as a result of the analysis presented.

Of the 99 cases, 73.33% of the "under observation" cases and 92.86% of the "serious-critical" cases were classified correctly for a total percent of 89.89% correct classifications. Additionally, Table 8 refers to the error measurement, but differentiating the weight for each class according to the criterion of the specialists. Thus, both performance measures corroborate that the SAPRIM system responds at the level of the human experts.

Table 7. Contingency table

EXPECTED CLASS/ SAPRIM'S PREDICTION	"UNDER OBSERVATION"	"SERIOUS-CRITICAL"
"UNDER OBSERVATION"	11	4
"SERIOUS-CRITICAL"	6	78

Table 8. Error considering different weights for each type of error

WEIGHT FROM THE ERROR WHEN CLASSIFYING AN "UNDER OBSERVATION" CASE AS "SERIOUS"	WEIGHT FROM THE ERROR WHEN CLASSIFYING A "SERIOUS" CASE AS "UNDER OBSERVATION"	ERROR
0.4	0.6	0.05

4 Conclusions

This paper has presented an automatic system for the prediction of mortality risk in pediatric patients, which uses Soft Computing techniques instead of traditional ones based on the score calculated from patient data. It is named SAPRIM (*Automated Predictor System of Infant Mortality Risk*). It is a classification problem since the medical investigations indicated that from 33 known patient's characteristics (predictive attributes), the patient evaluation (either the class "Under observation" or "Serious-critical") should be decided. Besides this, a memory of instances (the training set) from data contained in clinical histories of patients was available.

The machine learning tool WEKA was used for a comparative evaluation between classic classification algorithms and the variants of the hybrid model applied based on soft computing techniques. Several performance measures after applying 10-fold cross validation showed the best results in favor of ConFuCiuS. Additionally, SAPRIM uses linguistic terms similar to human experts, who can therefore understand the solution given. The new patient can be evaluated in the context of

patients with similar symptoms seen in the past, ranking the influence of each predictive attribute value on the prognosis obtained. The field evaluation corroborated that the SAPRIM system diagnoses at the level of the human experts.

References

1. Riesbeck, C., Schank, R.: Inside Case-Based reasoning. Lawrence Erlbaum Associates, New Jersey (1989)
2. Kolodner, J.: An introduction to case-based reasoning. Artificial Intelligence Review 6, 3–34 (1992)
3. Bergmann, R., Breen, S., Göker, M., Manago, M., Wess, S.: Developing IndustrialCBR Applications. In: Bergmann, R., Althoff, K.-D., Breen, S., Göker, M.H., Manago, M., Traphöner, R., Wess, S. (eds.) Developing Industrial Case-Based Reasoning Applications. LNCS (LNAI), vol. 1612. Springer, Heidelberg (2003)
4. Zadeh, L.A.: From Computing with Numbers to Computing with Words -From Manipulation of Measurements to Manipulation of Perceptions. In: Intelligent Systems and Soft Computing, pp. 3–40 (2000)
5. Pal, S.K., Shiu, S.C.K.: Foundations of soft case-based reasoning. Wiley series on intelligent systems. John Wiley & Sons Inc., Hoboken (2004)
6. Weber, R.O.: Fuzzy Set Theory and Uncertainty in Case-Based Reasoning. Engineering Intelligent Systems (2006)
7. Hüllermeier, E.: Case-Based Approximate Reasoning. Series B: Mathematical and Statistical Methods, vol. 44. Springer, Heidelberg (2007)
8. Zadeh, L.A.: Soft Computing and Fuzzy Logic. IEEE Software 11(6), 48–56 (1994)
9. Kuncheva, L., Steimann, F.: Fuzzy diagnosis. Artificial Intelligence in Medicine 16(2), 121–128 (1999)
10. Buisson, J.-C.: Approximate reasoning in computer-aided medical decision systems. In: Practical Applications of Fuzzy Technologies. Springer, Heidelberg (1999)
11. Pollack, M., Ruttimann, V., Getson, P.: Pediatric Risk of Mortality (PRIMS) score. Crit Care Med., 1011 (1988)
12. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)
13. Rodriguez, Y., Garcia, M.M., De Baets, B., Bello, R., Morell, C.: Extending a Hybrid CBR-ANN Model by Modeling Predictive Attributes using Fuzzy Sets. In: Dumke, R.R., Abran, A. (eds.) IWSM 2000. LNCS, vol. 2006, pp. 238–248. Springer, Heidelberg (2001)
14. Rodriguez, Y., Garcia, M.M., Baets, B.D., Morell, C., Bello, R.: A Connectionist Fuzzy Case-Based Reasoning Model. In: Gelbukh, A., Reyes-Garcia, C.A. (eds.) MICAI 2006. LNCS (LNAI), vol. 4293, pp. 176–185. Springer, Heidelberg (2006)
15. Rodriguez, Y., García, M.M., De Baets, B., Morell, C., Bello, R.: Integrating ANN and CBR using Fuzzy sets to develop Hybrid Systems. In: Hybrid Artificial Intelligence Systems. HAIS 2006 (2007)
16. García, M.M., Bello, R.: A model and its different applications to case-based reasoning, vol. 9(7), pp. 465–473 (1996)
17. Lacroix, J., Cotting, J.: Severity of illness and organ dysfunction scoring in children. Pediatr Crit Care Med., 126–134 (2005)
18. Mitchell, T.: The Need for Biases in Learning Generalizations. Readings in Machine Learning (1980)
19. Ruspini, E.: A new approach to clustering. Information and Control 15, 22–38 (1969)
20. le Cessie, S., van Houwelingen, J.C.: Ridge Estimators in Logistic Regression. Applied Statistics 41(1), 191–201 (1992)
21. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufman, San Mateo (1993)

Cell Formation with Alternative Routings and Capacity Considerations: A Hybrid Tabu Search Approach

Luiz Carlos A. Rodrigues and Tiago R. Weller

UTFPR, DAMEC/PPGEM, Av. Sete de Setembro 3165
80230-901 Curitiba, PR, Brazil
{lcar,weller}@utfpr.edu.br

Abstract. Cell formation considering the availability of alternative routings and the processing capacity of available machines is a challenging optimization problem, since it imposes the solution of a two step problem: *i*) assign machines to cells; and *ii*) assign manufacturing parts (routings) to cells. The objective is to minimize extra-cellular processing of tasks when routings of parts are selected. A hybrid tabu search approach is proposed to solve this cell formation problem. The proposed approach applies tabu search to assign machines to cells and branch-and-bound to select routings. In order to reduce the search space, a set of feasibility and pruning tests are proposed. A problem proposed and used at previous papers is used to compare the results of the proposed approach and the results presented by previous papers.

Keywords: Group Technology, Tabu Search, Branch-and-Bound.

1 Introduction

Production increases, product cost reductions, and work-in-process (WIP) inventory management improvements have been increasingly sought by industries. Despite the concepts and philosophies associated to Just-in-Time, which enforce cell formation, setting production cells can be a challenging task. The most used method in industry is to try to associate manufactured parts that are visually similar. The main drawback of this procedure is that cells can either overcome or misuse their production capacity. That is, it may not be an easy task to set sound and balanced cells by relying only on visual similarity of manufactured parts. Group Technology (GT) has been proposed to allow the formation of cells without taking into account the visual similarity of manufactured parts. GT seeks to identify groups of parts that share a group of machines [1]. GT usually relies on a machine-component incidence matrix to identify groups of machines and parts (or components) that can be grouped into cells. GT has evolved since its first proposition and an extensive literature on cell formation has been produced, indicating that several heuristic procedures have been proposed for this problem [2] [3] [4]. Some of the criteria for measuring the quality of cell formation are [2]: *i*) Minimization of extra-cellular operations; *ii*) minimization of the extra-cellular machine's load; and *iii*) minimization of the inter-cells production flow.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 482–491, 2008.

This paper is intended to solve the cell formation problem with alternative routings and production capacity restrictions. The objective is to minimize extra-cellular operations. But no matter the criteria used, the main difficulty of any procedure used for this problem is that the quality of a solution is measured only after machines are grouped. That is, this is a challenging problem because it is usually solved in two steps. First, machines are grouped into cells and then a routing is selected for each part. When a routing is selected to a part, this part can be associated to a cell with the smallest amount of extra-cellular tasks. This association permits to calculate the objective function.

At the present paper only a brief review related to the cell formation problem is presented. This review is intended to indicate that some heuristics have already been proposed to solve the cell formation problem with alternative routings and production capacity restrictions. A heuristic using linear programming was proposed by [5] to solve this problem. They iteratively solve the two steps of the cell formation problem as a two steps hill-climbing procedure. This same problem was also solved using a hybrid genetic algorithm [3] and simulated annealing [4] [6]. But at [4], simulated annealing was used only to solve the first step – machine grouping into cells – and branch-and-bound was used to solve the second step of the problem – associating parts to cells in accordance with the objective function. The hybrid genetic algorithm proposed by [3] uses a genetic algorithm for the machine grouping into cells. Then the association of parts to cells is performed using a Dantzig–Wolfe decomposition approach. Ant colony system was used to solve a quite similar problem [7], where the objective is to minimize the total cell load variation and the total intercellular moves.

To the authors' surprise, no paper was found applying tabu search to solve the proposed problem. Tabu search has been used by [8] and [9] to solve multi-objective cell formation problems. Both approaches disregard alternative routings and production capacity restrictions. Therefore the problem can be solved in one step because machine grouping into cells is enough to enable the objective function calculation. Although the problem solved by [8] and [9] is simpler than with alternative routings and production capacity restrictions, only some of the possible neighbor solutions are randomly generated at each iteration. Mixed integer programming, greedy and tabu search were used by [10] to solve the sustainable cell formation problem, where the routings of parts can change. The contribution of this paper is on discussing which machines to buy (or not) when a new cell is created, or when new parts are about to be manufactured. Only small problems are solved by [10]. Additionally, the reader can find extended reviews on the literature of the cell formation problem and its variations at [2], [3], and [4]. In order to solve the proposed problem, a hybrid Tabu search implementation is proposed and its results are compared to [4]. At the proposed hybrid Tabu search implementation, Tabu search is used to solve the first step of the problem and Branch-and-Bound is used to solve the second step. This paper is organized as follows. In section 2, the description and formulation of the cell formation problem with alternative routings and production capacity restrictions are presented. The proposed hybrid tabu search approach and a discussion of its contributions are presented at section 3. Computational results and conclusions are presented at sections 4 and 5, respectively.

2 The Cell Formation Problem

Table 1 presents the notation adopted for the modeling of the proposed cell formation problem. The proposed model for the manufacturing cell formation problem with alternative routings and production capacity restrictions is based on the following set of basic assumptions:

i. Production throughput is stable over time and identified as predetermined quantities of parts produced within a given planning horizon;
ii. Each part *i* is obtained after a sequence of O_k operations;
iii. A set of machines ($m = 1, \ldots, M$) is available to produce all parts and some of the machines are similar;
iv. Maximum number of machines allowed per cell ($c = 1, \ldots, C$) is NM.

Table 1. Problem notation

Parameters:

Cap_m	is the available processing capacity of machine m
d_p	is the demand of part p during the planning horizon
FM_{mr}	identifies the machine used after routing r is performed at machine m
MR_r	is the set of machines used by routing r
RM_m	is the set of routings used by machine m
RP_p	is the set of routings associated to part p
tp_{rm}	is the processing time of routing r at machine m
NM	is the maximum number of machines that can be allocated at a cell

Variables:

$$x_{mc} = \begin{cases} 1, & \text{if machine } m \text{ is allocated to cell } c, \\ 0, & \text{otherwise;} \end{cases}$$

$$y_{rp} = \begin{cases} 1, & \text{if routing } r \text{ is choosen for the production of part } p, \\ 0, & \text{otherwise.} \end{cases}$$

The mathematical model of this cell formation problem is as follows. Equation (1) is used when the objective of the problem is the minimization of extra-cellular operations. It is assumed that machine m precedes machine m' at the processing of routing r. This equation indicates that when a routing r is chosen for the processing of part p (indicated by $y_{rp} = 1$), then the presence of machines m (indicated by $x_{mc} = 1$) and m' (indicated by $x_{m'c} = 1$) at the same cell c is analyzed. Notice that a extra-cellular operation is identified only if $x_{mc} = 1$ and $x_{m'c} = 0$.

The studied problem is subject to the following constraints. Constraint (2) imposes that any part p must necessarily be associated to only one routing r. Constraint (3) imposes that any machine m must be associated to a cell c. Constraint (4) enforces

that allocation of parts to routings must take into account the processing capacity of all machines. Constraint (5) sets a limit to the number of machines allocated at a cell.

$$(Min)\ Z = \sum_{p} \sum_{\substack{r \\ r \in RP_p}} y_{rp} \sum_{\substack{m \\ m \in MR_r}} \sum_{c} x_{mc} \left(1 - \sum_{\substack{m' \\ m' = FM_{mr}}} x_{m'c} \right). \tag{1}$$

$$\sum_{\substack{r \\ r \in RP_p}} y_{rp} = 1 \qquad ,\forall\, p \tag{2}$$

$$\sum_{c} x_{mc} = 1 \qquad ,\forall\, m\,. \tag{3}$$

$$\sum_{p} d_p \sum_{\substack{r \\ r \in RP_p \\ r \in RM_m}} tp_{rm}\ y_{rp} \le Cap_m \qquad ,\forall\, m\,. \tag{4}$$

$$\sum_{m} x_{mc} \le NM \qquad ,\forall\, c\,. \tag{5}$$

Due to the computational complexity of this model and other models proposed at the literature by [3], [4], and [6], a hybrid tabu search approach is proposed to solve the cell formation problem with alternative routings and production capacity considerations.

3 Proposed Approach

A hybrid tabu search approach is proposed to solve the cell formation problem with alternative routings and capacity considerations. The proposed approach is divided into two steps that are solved iteratively. Both steps present pruning procedures that have not been seen by the authors at previous papers. At the first step, tabu search (TS) is used to perform machine allocation to cells, while respecting maximum amount of machines, *NM*, at any cell. At the second step, a branch-and-bound (BB) procedure is proposed to identify the routings that minimize extra-cellular processing of parts, while respecting the processing capacity of machines, Cap_m. The idea of using a hybrid TS procedure to solve the proposed cell formation problem came from the lack of literature using TS to solve cell formation problems. In fact, the available literature using TS to solve related problems, as at [8] and [9], have proposed procedures that does not explore the whole set of neighbor solutions. Instead, they randomly generate only some of the possible neighbor solutions. Therefore, it became clear that – in order to use TS efficiently – it is necessary to prune the search space, eliminating infeasible or not promising neighbor solutions. Additionally, it would be necessary to implement an efficient part-routing assignment procedure, which is solved using BB at this paper. Otherwise, the search can easily become time consuming.

3.1 First Step: Machine Allocation to Cells Using Tabu Search

This step of the proposed hybrid procedure is solved using TS [11]. This metaheuristic procedure explores possible solutions on the neighborhood of a current solution. It is meant to identify the best neighbor solution of a current solution using predefined movement strategies. Then the best neighbor solution replaces the current solution and the neighborhood search is restarted once again. Several memory structures may be adopted in order to foster the search and to avoid it from being trapped on regions of local optimal solutions. The reader may find many memory structures frequently found at TS implementations at [11]. But there is one memory structure – called the Tabu list – that is always found at TS. When a movement is performed and it provides the best neighbor solution, this movement becomes forbidden (or *tabu*) for *N* iterations. That is, this movement is inserted into the Tabu list in an attempt to avoid cycles of the algorithm. A movement placed at the Tabu list is only accepted if it provides the best solution found until that point of the search. This procedure is called the Aspiration Criteria. TS is characterized by two search situations. The first situation is called *Intensification*, when the search gets closer to a promising region and this region tends to be intensively searched. If it is identified that the search has become trapped on a region of local optima, a *diversification* is performed. Diversification is obtained generating a new initial solution or performing several consecutive movements that moves the search away from the region of local optima. The search is restarted after diversification.

The implemented TS algorithm is presented at Fig. 1. Diversification was not implemented at the proposed TS procedure because – at all tested instances – the search could reach the optimal solution without it. Initial solution is randomly generated. The current solution is identified as *s*, the best neighbor solution is identified as *s*' and the best overall solution is identified as s^{best}. Neighborhood search subroutine is called at line 4 of the main TS algorithm, indicated at Fig. 1.

As mentioned previously, in order to use TS efficiently it is necessary to prune the search space, eliminating infeasible or not promising neighbor solutions. Previous works on the proposed problem as [4], [6], [8], and [9] have used two movements: *i*) *insertion*, when a machine is moved from one cell to another; and *ii*) *swap*, when two machines exchange cells. Unlike previous works, only insertion moves were implemented at TS because swap moves could significantly increase the search space and because insertion seemed enough for a good performance of the implemented algorithm. Therefore, it was decided to test all feasible insertion moves, while at the TS implementations of [8] and [9] only some of the neighbor solutions where randomly generated. While generating neighbor solutions, three restrictions have been considered: *i*) insertion moves can only be made to cells with less than *NM* machines, as a feasibility restriction; *ii*) any cell should remain with a minimum amount of machines. This restriction was imposed as an attempt to avoid not promising neighbor solutions and minimum amount of two machines per cell was imposed; and *iii*) an insertion move of a machine *m* can only be made to another cell *c* if there is at least one machine related to *m* at cell *c*. Two machines are considered related if there is at least one routing where both machines are present. This pruning procedure is based on the fact that an insertion to a cell with no other related machines can not be better than the current solution. At the same time, avoiding unpromising solution can save search

time. A discussion is made of why using only insertion moves at section 4. Also, it is analyzed why the simplest structure of TS – indicated at Fig. 1 – was enough to solve the proposed problem.

```
1  Generate initial solution (s)
2  s^best = s
3  While stopping criteria is not satisfied, do:
4    Generate neighbors of s and select best neighbor(s')
5    If s' is better than s^best then:
6      s^best = s'
7    end if
8    Current solution is updated: s = s'
9  end while
10 Write s^best
```

Fig. 1. Implemented Tabu search algorithm

3.2 Second Step: Part-Routing Assignment Using Branch-and-Bound

At the proposed hybrid TS procedure, a BB search is performed whenever a feasible (due to an acceptable allocation of machines to cells) neighbor of *s* solution is produced. The objective of this step of the hybrid TS is to set part-routing assignments that minimize extra-cellular operations. But part-routing assignments must take into account the processing capacity of all machines. The challenge of BB is that, if not properly implemented, it can present an important computational burden. In order to perform part-routing assignments, an algorithm is proposed:

1. Since machines have been allocated to cells at the first step, identify the number of extra-cellular operations for each routing *r* of part *p*, as its assignment is tested at each cell *c*. Identify the smallest number of extra-cellular operations for each routing *r*, as $ECmin_r$, and the cell where it happened;
2. For all parts, sum the smallest $ECmin_r$ of each part *p*. This sum is used as the relaxed solution or lower bound of the root node of the BB;
3. For each part *p*, identify the difference between the two smallest $ECmin_r$, indicated as $diff_p$. Parts are ordered at a list called $EXPAND_n$ – where *n* will be used to identify node *n* at BB – according to the decreasing value of $diff_p$. The part with the biggest $diff_p$ is ordered first and the part with the smallest $diff_p$ is ordered last. Parts with only one possible routing assignment are not included on $EXPAND_n$ list, since their part-routing assignment is already set. The goal of $EXPAND_n$ list is to foster a depth-first BB;
4. For any part *p* with only one possible routing assignment *r*, recalculate the available capacity of any machine *m* associated to routing *r* (where $m \in MR_r$). This is done by subtracting the total processing time of routing *r* – considering the demand of part *p* – at machine *m* from the available capacity, Cap_m;
5. Perform BB search. At the root node, $EXPAND_{root}$ list contains all parts except for those with only one possible routing assignment. $EXPAND_n$ list is updated as branching of nodes is performed and each node *n* will have its own

$EXPAND_n$ list of parts not yet branched. As branching of the search tree is started at a root node, an ordered list of open nodes (partial solutions) called *OPEN* is used to guide the BB search. For minimization problems, *OPEN* list is ordered from the smallest (first node) to the greatest (last node) evaluation or objective function value that is associated to each node *n*. Initially, *OPEN* list is composed only of the root node. The lower bound of the BB is calculated as indicated on step 2 and the upper bound is set to $+\infty$. An in deep explanation of *OPEN* list is given at [12]. BB search is structured as follows:

i) Apply branching rule to the first node of *OPEN* list. This means that the first manufactured part *p* included on $EXPAND_n$ list will be branched. The set of routings associated to part *p*, RP_p, indicates the amount of new nodes that are created;
ii) Remove the first node from *OPEN* list and from $EXPAND_n$ list;
iii) Available capacity of machines, Cap_m, is updated at every new node. If the capacity of any machine is exceeded when a new node is created, this node is considered infeasible and it is eliminated from the search tree;
iv) Calculate the objective function value for each feasible new node associated to routing *r* of part *p*. Objective function value is given by summing the lower bound to Δ, where Δ is the difference between $ECmin_r$ for routing *r* and the smallest $ECmin_r$ for part *p*. Notice that $\Delta \geq 0$;
v) If a feasible open node (partial solution) was created at step (5.*i*), it is ordered and included on *OPEN* list;
vi) If $EXPAND_n$ list is empty, new complete solutions (closed nodes) were identified. If a new solution is feasible and it is smaller than the upper bound, the upper bound is updated to this solution's objective function value. This solution is saved as the best solution yet (*s'*);
vii) Perform fathoming procedure. Any node is eliminated from the *OPEN* list if its objective function value is greater than the upper bound;
viii) BB search is ended if *OPEN* list is empty. Otherwise, return to step 5.*i*;
ix) Return **s'** as the best neighbor solution.

The greatest contribution of the presented BB procedure is on proposing the use of $EXPAND_n$ list. It is intended foster as much as possible a depth-first search. Therefore there seems to be a significant pruning of the search space of BB, as indicated by the proposed approach processing times.

4 Computational Results

The proposed approach has been run on a Pentium IV PC with CPU of 3.00 GHz and RAM memory of 2.00 GB. This approach was implemented on a Visual Studio 2005 environment. The example published in [5] was used to test the proposed approach, since it was also used by [4] to test their hybrid SA approach. The results of the proposed approach are compared to those of [4]. In this example, 20 parts, 51 routings and 20 machines have to be grouped into 5 cells. Initial solution was obtained randomly.

Table 2 indicates the results presented by [4] in number of iterations. In order to test SA, the two parameters were changed at each time: initial temperature (*C*) and swap probability (*pp*). That is, a neighbor is created from a solution by using swap with probability *pp* and by using insertion with probability (1 - *pp*). For each set of parameters *C* and *pp*, 25 runs have been performed. The optimal solution given in [4] has been found in 218 iterations in the best case (where $C = 10$ and swap probability $pp = 0.5$). Tests on the algorithm presented by [4] have been consistent with those presented at table 2, since the standard deviation from the average (when the average was smaller than 600) has ranged from 199.94 to 333.76. Table 3 presents a comparison between the results after 100 runs of the proposed approach – hybrid TS – and the SA approach of [4]. The only parameter that was set for the hybrid TS was the size of the Tabu list, which has been set to 5 iterations due to the size of the proposed instance. For the problem proposed by [5], it can be seen that the proposed approach – hybrid TS – outperforms the best results presented at [4]. Worst CPU times were used to compare the processing times of both approaches because of the significant statistical variance presented by SA. In fact, an important drawback of SA was on its statistical variance.

Table 2. Number of iterations required to find the optimal solution [4]

	C				
pp	1	5	10	15	20
0	2000	397	450	743	1538
0.25	1031	556	321	308	1622
0.50	775	319	218	949	1100
0.75	764	280	671	1861	2000
1.00	2000	2000	2000	2000	2000

Table 3. Number of iterations required to find the optimal solution using hybrid TS and SA

Approach	Number of iterations	Worst CPU time
Hybrid TS	12.7 ± 3.45	2 seconds
Hybrid SA [4]	218 ± 199.94	5 seconds

It is believed that the main reason for the use of SA – as seen on literature review – for the proposed problem is that it has to be solved on two steps. Additionally, the fact that BB is performed at the second step of the proposed problem has played against approaches like TS and GA, which explore several neighbor solutions at each iteration. Due to these presumptions of the authors, the proposed hybrid TS performed only insertion moves – at its first step – in an attempt to restrain the space of neighbor solutions. Notice three rules where used to search neighbor solutions at the first step of hybrid TS, but only the first one was a problem restriction. The two pruning rules that were added to first step contributed to reduce the space of neighbor solutions. The second important contribution of this paper was on incorporating $EXPAND_n$ list to BB, which made the algorithm processing time reasonable.

A comment on the used example, proposed by [5], is necessary. It was the only one found for the proposed problem and it had been used at previous comparisons among different approaches. The literature on cell formation problem indicates the use of problems of similar size. Despite this, the used example was considered small. Due to this, hybrid TS converged quite fast to the optimal solution, making *diversification* unnecessary for this example. Since the results have been promising, bigger examples will be used in the future.

5 Conclusion

This paper is focused on the solution of the cell formation problem with alternative routings and capacity considerations. A new mathematical model was proposed for this problem. This is a challenging problem that requires two steps to solve one iteration using heuristics: *i*) machine allocation to cells; and *ii*) objective function calculation. Due to this, only a limited range of heuristics has been proposed to solve it. A hybrid TS approach has been presented with two important contributions: two pruning rules for the first step of the problem and a fast BB procedure to solve the second step of the problem. Hybrid TS results have outperformed those of a hybrid SA presented by [4]. These are the first results of the proposed hybrid TS approach. Examples of bigger size will be proposed in the future, making it possible to test other TS features like *diversification*, reactive TS [11], or when to use swap moves (for example, at *diversification*), etc. Additionally, other approaches like GA can be tested in the future, using the proposed BB procedure to solve objective function calculation. Finally, the author would kindly ask other authors to indicate their results using both average and standard deviation (or variance) values.

References

1. Burbidge, J.L.: An Introduction to Group Technology. John Wiley, New York (1975)
2. Irani, S.A.: Handbook of Cellular Manufacturing Systems. John Wiley, New York (1999)
3. Nsakanda, A.L., Diaby, M., Price, W.L.: Hybrid Genetic Approach for Solving Large-Scale Capacitated Cell Formation Problems with Multiple Routings. European J. of Operational Research 171, 1051–1070 (2006)
4. Caux, C., Bruniaux, R., Pierreval, H.: Cell Formation with Alternative Process Plans and Machine Capacity Constraints: A New Combined Approach. Int. J. Production Economics 64, 279–284 (2000)
5. Nagi, R., Harhalakis, G., Proth, J.M.: Multiple Routeings and Capacity Considerations in Group Technology Applications. Int. J. Production Research 28, 2243–2257 (1990)
6. Xambre, A.R., Vilarinho, P.M.: A Simulated Annealing Approach for Manufacturing Cell Formation with Multiple Identical Machines. European J. of Operational Research 151, 434–446 (2003)
7. Prabhaharan, G., Muruganandam, A., Asokan, P., Girish, B.S.: Machine Cell Formation for Cellular Manufacturing Systems Using an Ant Colony System Approach. Int. J. Adv. Manuf. Technol. 25, 1013–1019 (2005)
8. Lei, D., Wu, Z.: Tabu Search-Based Approach to Multi-Objective Machine-Part Cell Formation. Int. J. Production Research 43, 5241–5252 (2005)

9. Lei, D., Wu, Z.: Tabu Search for Multi-Criteria Manufacturing Cell Design. Int. J. Adv. Manuf. Technol. 28, 950–956 (2006)
10. Foulds, L.R., French, A.P., Wilson, J.M.: The Sustainable Cell Formation Problem: Manufacturing Cell Creation with Machine Modification Costs. Computers & Operations Research 33, 1010–1032 (2006)
11. Glover, F., Laguna, M.: Tabu Search. Kluwer Academic, Norwell (1997)
12. Pearl, J.: Heuristics: intelligent search strategies for computer problem solving. Addison-Wesley, Reading (1984)

Fault Diagnosis of a Vehicle with Soft Computing Methods

Juan Pablo Nieto González[1], Luis E. Garza Castañón[1], Abdelhamid Rabhi[2], and Ahmed El Hajjaji[2]

[1] Department of Mechatronics and Automation, ITESM Monterrey Campus
Av. Eugenio Garza Sada Sur No. 2501
Monterrey, N.L. 64,489 México
juan.pablo.nieto@itesm.mx,legarza@itesm.mx
[2] Laboratoire Modélisation, Information et Systémes
Université de Picardie Jules Verne (UPJV)
33 Rue Saint Leu
80039 Amiens, France
abdelhamid.rabhi@u-picardie.fr,ahmed.hajjaji@u-picardie.fr

Abstract. The main goals of an on-board fault detection and diagnosis system in a vehicle are mainly to avoid damage to the vehicle and prevent dangerous situations for occupants. These goals can achieved by triggering correcting actions or at least by warning the driver. However, a vehicle is a complex system where monitoring and diagnosis is challenging due to inherent uncertainty caused by noisy sensors, measurements and non modeled dynamics. In this work a new approach is presented to perform online diagnosis in a vehicle, which is able to deal with measurement noise presence. Our method analyzes with fuzzy logic and neural networks the residuals obtained by the comparison between sensors measurements and the output of a mathematical model of vehicle dynamics. Experiments shows promising results when faults are induced during different vehicle maneuvers simulations.

Keywords: Vehicle Dynamics, Fault Detection, Fault Diagnosis, Soft Computing, Fuzzy Logic, Competitive Neural Network.

1 Introduction

Vehicle fault diagnosis relies on the processing of signals coming from sensors, which have dynamic ranges in magnitude, oscillation, frequency, slope, derivative, etc. Also noise can be present in these sensor measurements, making extremely difficult to develop a complete diagnostic tool that can fully answer all the questions related to automotive engineering faults.

In general, fault diagnosis systems have to deal with some sort of uncertainty coming from the non modeled dynamics and noisy measurements. For this reason, a threshold has to be used to declare the existence of a fault. However the computation of thresholds has to be done carefully to perform the rejection of

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 492–502, 2008.

noisy signals and allow the detection of real faults. In the past, different combination of techniques have been applied to deal with this problem. Nevertheless, in general they can not deal with high measurement noise presence without diminish their performance of diagnosis. For instance, authors in [1] use a bank of observers combined with estimators, then the residuals are compared using a sequential test of Wald, but it is pointed out that this method has limitations in presence of high measurement noise. The approach presented in [2] applies a dynamic Bayesian network on the residuals generated by multiple simple models of a vehicle, but propagating the dynamic Bayesian network requires calculating and storing an exponentially large probability distribution giving thus good results only for small number of faults. The framework presented in [3] is based on [2], it takes into account noise presence, but with the assumption that the Bayesian network must sample the residuals slower than the noise dynamics.

Related works to ours are reported in [4] and [5]. In the first paper, they propose two methods to implement a driving condition detector. The first method is based on a characteristic velocity stability indicator (CVSI) generated from a two degrees of freedom model of lateral dynamics. The CVSI is used to distinguish between different driving and stability conditions, such as understeering, oversteering and neutralsteering. This method uses fixed thresholds. The second method uses fuzzy thresholds in combination with a calculated stability condition (based on characteristic velocity and longitudinal velocity), steering wheel angle and yaw rate. In [5] a fuzzy system is proposed to perform tolerance to noisy measurements, but they just take into account sensor faults greater than 10% during normal operation and non-critical driving situation. In this work, fault symptom relationships between residuals are derived for each type of fault. These relationships are specified by taking into account qualitative residuals behavior such as: decrease, increase, don't care and no deflection.

In this work we propose a fault detection method which combines a vehicle model-based method and soft computing techniques applied over the residuals to carry out the diagnosis. Residuals are the values resulting from the discrepancies between the observed behavior and the prediction made by the vehicle model used. This combination of techniques is done in order to reduce the number of false alarms in the diagnosis process as well as to deal with the presence of high measurement noise. The framework is divided in two phases. In the first phase the vehicle model is used to obtain a set of residuals. In the second phase a vector is formed with the residuals and then, using fuzzy logic, a comparison against their normal operation limits is made to look for those residuals standing out of limits. After that, a competitive neural network classifies the output given by the fuzzy system, indicating in this way which fault is present on the vehicle. Soft computing techniques are used because they are more suitable to handle flexible thresholds for each variable being monitored, and this allows us to take a decision with the minimum number of false alarms in a relatively easy and fast way. Thus, the fuzzy logic system gives us the chance to distinguish between the presence of noise or outliers as it gives more relaxed thresholds than other approaches, and the competitive neural network locates the faulty variable. The

Table 1. Nomenclature

Variable	Definition
v_x	longitudinal velocity of the center of gravity
v_y	lateral velocity of the center of gravity
$\dot{\psi}$	the yaw velocity
ω_{ij}	wheel rotational velocity
C_i	the motor couple applied at wheel i
F_{xi}	is the longitudinal force applied at wheel i
F_{yij}	lateral force applied at each wheel i
J_r	inertia of the tire
R_ω	radius of the tire
T_i	the braking couple applied at wheel i
$r_{1,2}$	distance between gravity center and front (rear) axis
$p_{f,r}$	is the front (rear) half gauge
m	the total mass of the vehicle
δ_f	the front wheel steer angle (rear wheel $\delta_f = 0$)
J_z	yaw moment of inertia

organization of the paper is as follows: section 2 shows the vehicle model used in the proposal. Section 3 gives the framework general description. Section 4 shows the framework performance with three simulations examples taking into account different noise magnitudes. Section 5 gives the conclusions of this work.

2 Vehicle Model

The generation of residuals in this work is based on a complex 7 degrees of freedom vehicle model, which includes longitudinal and lateral motions, yaw motion and the rotational dynamics of the four wheels. The nomenclature used in this work is shown in table 1.

The equations of motion are given as:

$$\begin{aligned} M\dot{v}_x = {} & \dot{\psi}V_y + \frac{1}{m}[(F_{xfl} + F_{xfr}).cos(\delta_f) \\ & - (F_{yfl} + F_{yfr}).sin(\delta_f) + (F_{xrl} + F_{xrr})] \end{aligned} \tag{1}$$

$$\begin{aligned} M\dot{v}_y = {} & -\dot{\psi}V_x + \frac{1}{m}[(F_{xfl} + F_{xfr}).sin(\delta_f) \\ & + (F_{yfl} + F_{yfr}).cos(\delta_f) + (F_{yrl} + F_{yrr})] \end{aligned} \tag{2}$$

$$\begin{aligned} \dot{\psi} = {} & \frac{1}{J_z}[R_1(F_{xfl} + F_{xfr})sin(\delta_f) + R_1(F_{yfl} + F_{yfr})cos(\delta_f) \\ & - r_2(F_{yrl} + F_{yrr}) - p_f(F_{xfl} - F_{xfr})cos(\delta_f) \\ & + p_f(F_{yfl} - F_{yfr})sin(\delta_f) - p_r(F_{xr2l} - F_{xrr})] \end{aligned} \tag{3}$$

The subscripts f and r refer to front and rear wheels respectively. The rotation motion of the wheels can be added. Figure 1 shows the coordinate system fixed to the center of gravity, from which Newton laws were applied to obtain the above equations.

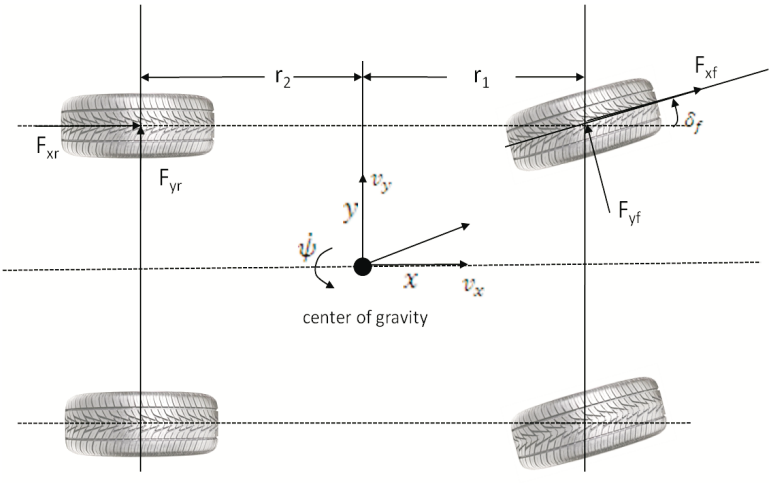


Fig. 1. Scheme for modeling the longitudinal and lateral motions of a vehicle

The wheel angular motion is given by:

$$\dot{\omega}_{fl} = \frac{1}{J_r}(-T_{f_l} - R_\omega F_{xf1}) \tag{4}$$

$$\dot{\omega}_{fr} = \frac{1}{J_r}(-T_{fr} - R_\omega F_{xf2}) \tag{5}$$

$$\dot{\omega}_{rl} = \frac{1}{J_r}(C_{rl} - T_{rl} - R_\omega F_{xr1}) \tag{6}$$

$$\dot{\omega}_{rr} = \frac{1}{J_r}(C_{rr} - T_{rr} - R_\omega F_{xr2}) \tag{7}$$

PAC 2002 [6], the latest version of well-known Magic Formula has been used for the plant model. This equation represents a semi-empirical model to calculate steady state tyre force and moment characteristics. The general form (sine version) of the formula is given as follows:

$$Y(x) = D\sin[C\arctan\{Bx - E(Bx - \arctan(Bx))\}] \tag{8}$$

where $Y(x)$ is either F_x with x the longitudinal slip ratio λ or F_y with x the lateral slip angle α.

The parameters in the Magic Formula depend on the type of the tire and road conditions. These parameters can be derived from experimental data obtained from tests.

3 Framework Description

During the first phase of diagnosis, residuals are generated by making a comparison between sensor measurements and model calculations. After that, a second phase based on fuzzy logic, looks for those residuals out of normal operation limits. Then, a competitive neural network classifies each output of the fuzzy system to know the actual operation mode of a system.

The general fault detection and diagnosis framework is shown in figure 2. First a data set of the residuals coming from the normal operation of the system is required. Then the minimum, maximum and statistical mean values for each of these residuals are obtained to design the rules of the fuzzy system. They will be used to verify if a residual is out of its normal operation limits. In this work, we use the gaussian membership function for the fuzzy system.

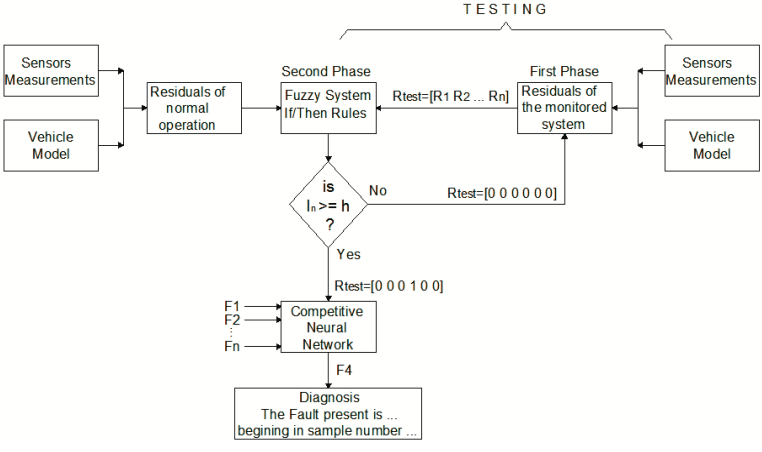


Fig. 2. General fault detection and diagnosis framework

The set of fuzzy rules are stated in this form:

- **If** value is inside limits **then** R_n is zero
- **If** value is outside limits **then** R_n is one

Where R_n is the residual obtained from the comparison between the n^{th} output of the model and the measurement of the n^{th} sensor.

The first phase delivers the residuals in a vector of real numbers R_{test} of dimension $1 \times n$:

$$R_{test} = [R_1, ..., R_n] \tag{9}$$

The R_{test} vector is analyzed by the fuzzy system to obtain the residuals out of limits. If this is the case, the fuzzy system assigns this residual a value of

one, otherwise, is indicated with a zero. In order to detect false alarms, an index must be defined to monitor the number of consecutive points, given by the fuzzy system, that lay out of their given limits. In this way we can distinguish between noise presence or the existence of a real fault. Thus, this index I_n will count the consecutive number of times or points the residual R_n has out of limits, in such way that when I_n reaches h, that is a constant number set by the user, a binary R^b_{test} vector will be formed as that shown in equation (10). Here, it is shown for instance, the R^b_{test} formed when the fourth residual R_4 of an R_{test} vector formed by six residuals, has reached h consecutive points out of its limits.

$$R^b_{test} = [0 \ 0 \ 0 \ 1 \ 0 \ 0] \tag{10}$$

This vector is the input of a competitive neural network classifier. For instance, the vector shown in (10) could indicate a fault in the right rear wheel sensor. As this process is carried out over each of the R_n residuals forming the R_{test} vector, it is possible to know in which time or sample number a fault started.

The algorithm of the second phase could be summarized as follows:

1. Obtain a set of residuals from a normal operation data set.
2. Obtain the minimum, maximum and mean values for each residual of the set of normal operation data from step **(1)**
3. Build the fuzzy system for each residual of the system being monitored according with the values obtained in step **(2)** with the **if/then** rules of the form :
   - **If** value is inside limits **then** R_n is zero
   - **If** value is outside limits **then** R_n is one
4. Define an index I_n for each residual in order to distinguish between noise presence and a fault present. **If** $(I_n \geq h)$ **Then** a fault is present on R_n **Else** measurement noise is present on R_n.
   h is the threshold of consecutive points out of the limit allowed before setting an alarm of a possible fault presence, and it is set by the user.
5. Learn a competitive neural network with the different fault signatures in order to classify the output of the fuzzy system.

Thus, when new residuals arrives, the diagnostic system will work as described below:

6. Build a residuals test vector of the form
   $R_{test} = [R_1, ..., R_n]$
7. Take the R_{test} vector as the input of the fuzzy system. The output of it is a R^b_{test} binary vector that will have a **0** if the residual is inside its normal operation limits or a **1** if it is out of them.
8. Verify the index of each residual I_n. **If** the points laying out of limits are consecutive **Then** increment the index I_n and **Go** to step **(9) Else** reset I_n to zero.
9. **If** $I_n \geq h$ **Then** stop the monitoring process and **Go** to step **(10) Else** return to step **(6)**.
10. Set the R^b_{test} vector obtained in step **(7)** as the input of the competitive neural network previously trained. Give which fault is present and its location.

4 Case Study

This section gives some results in order to test and validate the proposed method. We have performed several experiments, where the sensor measurements were generated by using the vehicle simulator VEDYNA [8]. The set of residuals under consideration are:

- **R1** front left wheel rotational velocity
- **R2** front right wheel rotational velocity
- **R3** rear left wheel rotational velocity
- **R4** rear right wheel rotational velocity
- **R5** vehicle longitudinal velocity
- **R6** yaw rate

We consider two different vehicle maneuvers, shown in figure 3: *chicane*, which is a double change of lane maneuver, and *slalom*, which is a maneuver to sort aligned cones. We generate three different databases:

1. Normal operation residuals for *chicane* database
2. Fault present in residual R1 *chicane* database
3. Fault present in residual R4 *slalom* database

Fig. 4 shows the output of the fuzzy system for the six residuals when vehicle works in normal operation for *chicane* database.

As mentioned above, the index defined in step **(d)** of the algorithm is used to distinguish between noise or fault present on a system. When a system has several points out of their limits, without reaching h consecutive points, they are considered as measurement noise presence.

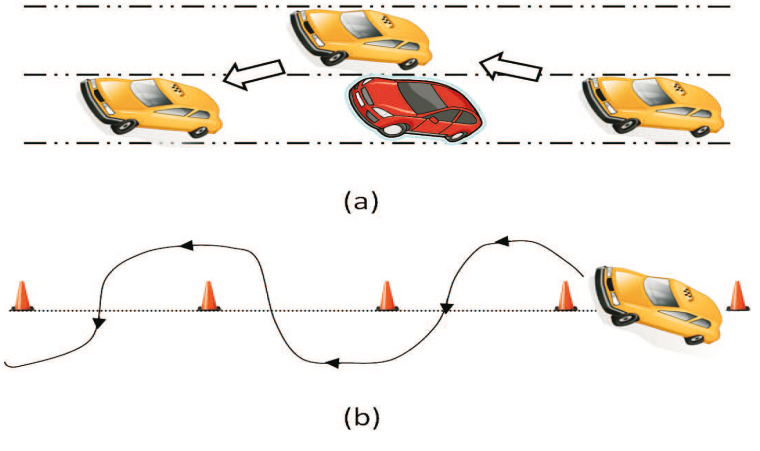


Fig. 3. Car maneuvers simulated in software VEDYNA. (a) *chicane* or double change of lane, (b) *slalom* or sorting aligned cones.

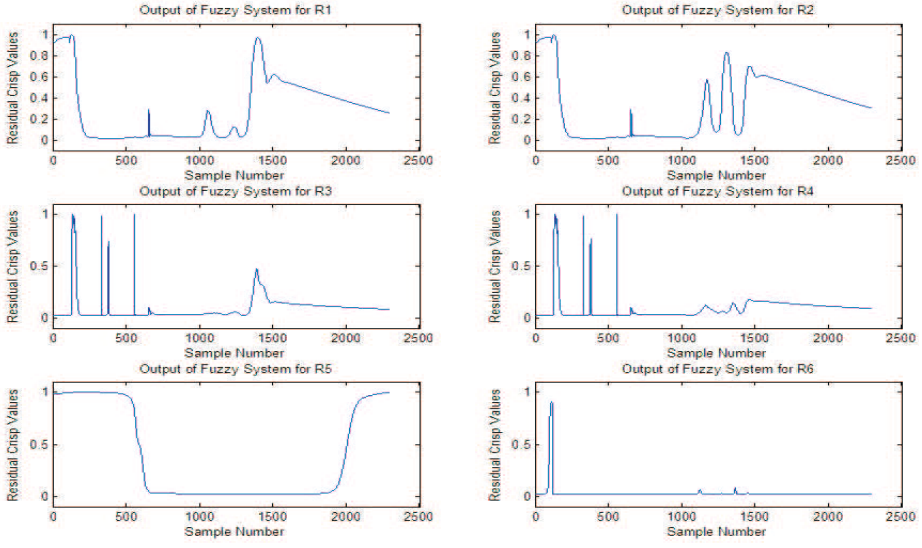


Fig. 4. Defuzzyfied output of the fuzzy system for normal operation residuals for *chicane* database

When there is a fault present on a certain residual, the output of the fuzzy system is taken as the input of the competitive neural network in order to determine the type of fault. As this process is carried over each residual, when a fault exists, its location over time is given immediately. After carry out several simulation tests it has been found out, as shown in table 2, that the best performance of the proposal is reached when $h = 10$.

Fig. 5 and Fig. 6 show the results of the proposal when a fault exists on R_1 and R_4 on *chicane* and *slalom* databases respectively. In both cases, the index used to distinguish between noise or fault was $h = 10$.

The results given by the fuzzy system when a fault is present in R_1 *chicane* database are explained as follows:

- There are 10 points out of the limits for residual R_1 and their locations are samples number 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1208, 1209 and 1210.
- There are 0 points out of the limits for the rest of the residuals.

As it was expected, just residual R_1 has points out of its limits because it was a simulation without measurement noise presence. In addition to this it could be noticed that the fault is found out when having $h = 10$ consecutive points out of the limits of R_1. Note that fault starts at sample 1201 that is the first of the ten consecutive points.

The results for fault present in R_4 *slalom* database simulation are as follows:

- There are 3 points out of the limits for residual R_2 and their locations are samples number 490, 491 and 492.

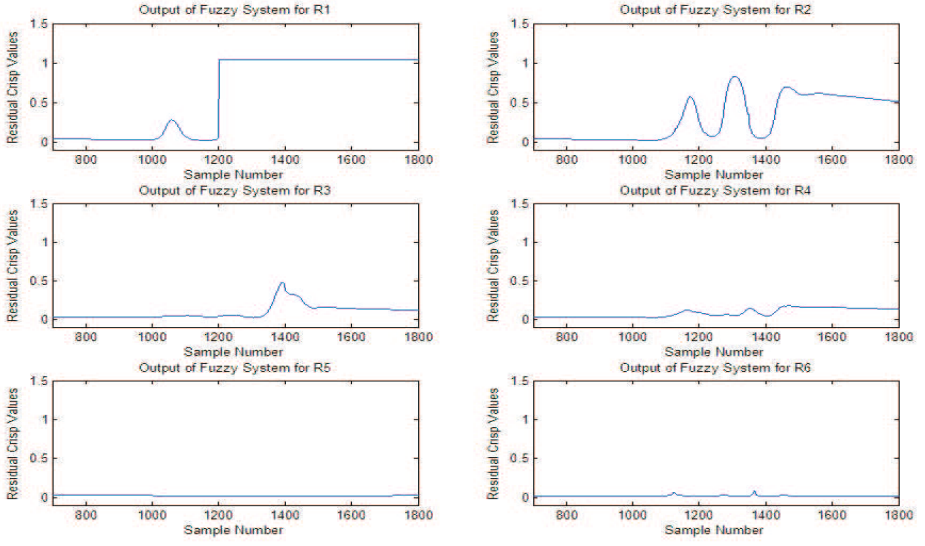


Fig. 5. Fault detection for a simulation of *chicane* data base with fault present in R_1 without noise and using $h = 10$

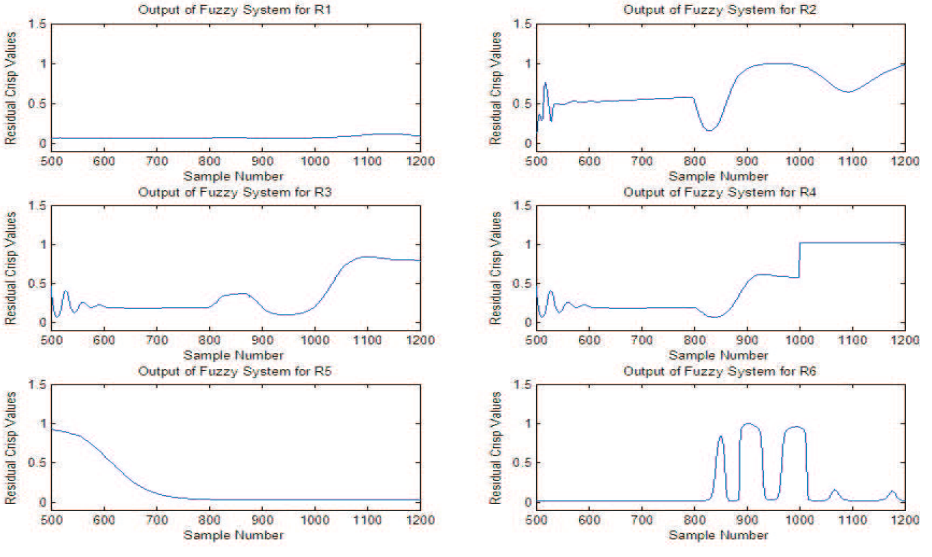


Fig. 6. Fault detection for a simulation of *slalom* data base with fault present in R_4 without noise and using $h = 10$

- There are 10 points out of the limits for residual R_4 and their locations are samples number 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008 and 1009.
- There are 0 points out of the limits for the rest of the residuals.

Table 2. Performance of detection for different magnitudes of measurement noise present in all the residuals simultaneously for different h values

Noise Magnitude	$h = 10$	$h = 7$	$h = 5$	$h = 3$
0.0001	100%	100%	100%	85%
0.001	100%	100%	90%	50%
0.01	100%	95%	60%	20%
0.1	100%	70%	25%	5%

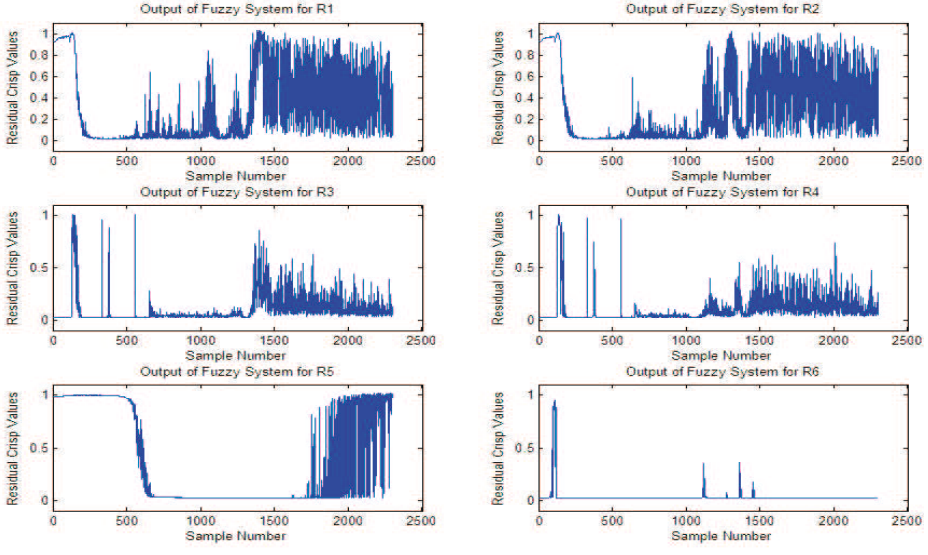


Fig. 7. Presence of measurement noise of 0.01 magnitude on normal operation residuals and $h = 10$

As R_2 has 3 consecutive points out of the limits, they could be taken as outliers (Fig. 6). But as it was expected, just residual R_4 has $h = 10$ consecutive points out of its limits indicating thus the presence of a fault on it. Note that fault starts at sample 1000 that is the first of the ten consecutives points.

4.1 Measurement Noise

Table 2 shows the simulations results when different measurement noise magnitudes are present, in all the residuals simultaneously, for different quantities of consecutive points (h). This table shows that, when there is measurement noise presence in the system, an index of $h = 10$ is preferred to be used as it could distinguish between noise and fault present in all the simulations that were carried out.

The performance percentages shown, were obtained from 20 simulations with 2300 samples each of normal operation for each pair formed with a noise magnitude and an h value. Giving thus a total of 320 simulations.

Fig. 7 depicts the behavior of the outputs from the fuzzy system for normal operation when there is measurement noise present of magnitude 0.01 and $h = 10$ in all the residuals simultaneously. In this case, as there were not $h = 10$ consecutive points out of the limits in any of the residuals, all the nonconsecutive points out of the limits are considered as measurement noise present.

5 Conclusions

This paper has presented an online fault detection framework for a vehicle, based on the processing of residuals with fuzzy logic and neural networks. The main advantages of this approach is that it diminishes the number of false alarms in the diagnosis process, and also deals with high measurement noise presence on sensors.

This framework is divided in two phases: the first phase uses a model based technique to obtain the residuals between the output of the model and the measurements. The second phase process the residuals with soft computing techniques. These are a combination of fuzzy logic and a competitive neural network. The implementation of an index, that grows up when a constant number of consecutive points is out of the limits given by the fuzzy system, allows us to distinguish between measurement noise or a real fault present in the system. A competitive neural network is used to classify the outputs of the fuzzy system in order to carry out the diagnosis. As the process diagnoses each residual individually, this approach detects which residual is in faulty mode, as well as the time when a fault occurred.

References

1. Zbiri, N., Rabhi, A., Sirdi, N.K.M.: Analytical Redundancy Techniques for Faults in Vehicle. In: Mediterranean Conference on Control and Automation, Limassol, Cyprus, June 27-29 (2005)
2. Schwall, M., Gerdes, J.: A Probabilistic Approach to Residual Processing for Vehicle Fault Detection. In: Proceedings of the American Controls Conference, pp. 2552–2557 (2002)
3. Schwall, M.L., Gerdes, J.C., Bäker, B., Forchert, T.: A Probabilistic Vehicle Diagnostic System Using Multiple Models. American Association for Artificial Intelligence, 123–128 (2003)
4. Börner, M., Isermann, R.: Model-based Detection for Critical Driving Situations with Fuzzy Logic Decision Making. Control Engineering Practice 14, 527–536 (2006)
5. Fischer, D., Börner, M., Schmitt, J., Isermann, R.: Fault Detection for lateral and vertical vehicle dynamics. Control Engineering Practice 15, 315–324 (2007)
6. Pacejka, H.B.: Tyre and vehicle dynamics. Butterworth Heinemann, Oxford (2002)
7. Tsoukalas, H., Uhrig, E.: Fuzzy and Neural Approaches in Engineering. Wiley Interscience, Chichester (1997)
8. Simulator VE-DYNA, http://www.tesis.de/index.php

Hybridization of PSO and a Discrete Position Update Scheme Techniques for Manufacturing Cell Design

Orlando Duran, Nibaldo Rodriguez,
and Luiz Airton Consalter

Pontificia Universidad Catolica de Valparaiso
Valparaiso
Chile
Universidade de Passo Fundo
Passo Fundo (RS)
Brasil
{orlando.duran,nibaldo.rodriguez}@ucv.cl;lac@upf.br

Abstract. This paper proposes an hybrid algorithm for Manufacturing Cell Formation. The two techniques that are combined to address this problem correspond to Particle Swarm Optimization (PSO) and a Data Mining Clustering application. The criterion used to group the machines in cells is based on the minimization of inter-cell movements. A maximum cell size is imposed and the number of cell is parameterizable. Some published exact results have been used as benchmarks to assess the proposed algorithm. The computational results show that the proposed algorithm is able to find the optimal solutions on almost all instances with low variability and stability.

Keywords: Manufacturing cells, machine grouping, particle swarm optimization.

1 Introduction

Cellular Manufacturing is an organizational approach based on Group Technology (GT). Cellular manufacturing aims to divide the plant into a certain number of cells. Each cell contains machines that process similar types or families of products. Manufacturing cells have been shown to provide considerable cost and productivity benefits to practical manufacturing environments. Other considerable number of benefits to be gained by grouping machines into cells can be found in literature [1], [3].

The major issue in the design of manufacturing cells is the identification of machine and component groups. This identification process requires an effective approach to form part families so that similarity within a part family can be maximized. According to Selim et al. [1] clustering analysis is the most frequently used method for MC design. However, and because of cellular formation problem

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 503–512, 2008.

(CFP) is a NP-complete problem [2], there is still the challenge of creating an efficient clustering method.

This paper deals with the use of a discrete particle swarm optimization algorithm [13] to effectively perform machine grouping into manufacturing cells. The remaining of the paper is organized as follows. In Section 2, we discuss related works concerning machine grouping and particle swarm optimization techniques. In Section 3, we describe the optimization model. Experimental results are presented in Section 4. Finally, in Section 5 we give some conclusions and lines for future research.

2 Background

The part-machine grouping problem was formally defined by [3] and can be stated as follows: given a set of machines, a set of parts, and a part-machine incidence matrix; assign parts and machines to a fixed number of part families and associated manufacturing cells, so that the cell coupling (measured by the number of out-of-cell operations or intercellular movements) is minimized and each cell does not contain more than a specified maximum number of machines. Cell coupling occurs due to parts requiring machines that belong to a cell different to its own. If a part is assigned to a given manufacturing cell, we call any operation included in its part route sheet which has to be performed outside the assigned cell an "exceptional element". For a given cell configuration, the degree of interaction between cells can be measured by the number of exceptional elements.

The attempts to solve the part-machine grouping problem can be classified in two groups. The first group consist of algorithms that attempt to determine the optimal solution. But, perhaps, the most successful approximate methods have been metaheuristic. Dimopoulos [4] examines the impact of the fast-growing evolutionary multiobjective optimization field in the area of production research. A considerable number of significant applications are reported for a wide range of relevant optimization problems among them we highlight the area of machine grouping or also referred as cell formation problem.

In the last years different metaheuristic methods have been used to solve the cell formation problem in group technology, Boctor [3] and Chen and Srivastava [4] used Simulated Annealing. Venugopal and Narendran [7] presented the first attempt to solve a cell-formation problem with the help of an evolutionary computation algorithm. They also proposed the first evolutionary optimization algorithm that explicitly considered the multiobjective version of the problem. Gupta et al. [8] employed the same genetic representation of solutions as Venugopal and Narendran but followed a different multiobjective optimization approach for the simultaneous minimization of the total number of intercell - intracell moves and within-cell load variation. Aljaber et al. [9] and Lozano et al. [10] used Tabu Search. More recently the paper of Andres and Lozano [11] presents the first particle swarm optimization (PSO) algorithm designed to the cell formation problem and to evaluate the ability of the algorithm to reach

optimal solutions. As an alternative to that work we propose here the utilization of a DPSO method for cell formation problem using the concept of proportional likelihoods. According to [13] the notion of proportional likelihood and the notion of velocity used in standard PSO are somewhat similar. This concept constitutes the main difference between this work and the ones cited before.

3 Particle Swarm Optimization

PSO is an evolutionary computation (EC) method inspired by flocking birds. This population based stochastic optimization technique was developed by Eberhart and Kennedy [10] and it was applied to many different areas including manufacturing. Although in the last years different metaheuristic methods have been used to solve the cell formation problem in group technology.

PSO is initialized with a population of random solutions and this initial population evolves over generations to find optima. However, in PSO, each particle in population has a velocity, which enables them to fly through the problem space instead of dying and mutations. Therefore, each particle is represented by a position and a velocity. Modification of the position of a particle is performed by using its previous position information and its current velocity. Each particle knows its best position (personal best) so far and the best position achieved in the group (group best) among all personal bests. These principles can be formulated as:

$$v_i^{n+1} = wv_i^n + c_1 r^n (p_i^n - x_i^n) + c_2 r_2^n (p_g^n - x_i^n) \tag{1}$$

$$x_i^{k+1} = x_i^k + v_i^{n+1} \tag{2}$$

where w is inertia weight; c_1 and c_2 are two positive constants, called cognitive and social parameter respectively; i=1, 2, , m, and m is the size of the swarm; r1, r2 are random numbers, uniformly distributed in [0,1]; and n=1, 2, , N, denotes the iteration number, N is the maximum allowable iteration number.

The first term on the right hand side of Eq. (1) is the previous velocity of the particle, which enables it to fly in search space. The second and third terms are used to change the velocity of the agent according to pbest and gbest. Generally speaking, the set of rules that govern PSO are: evaluate, compare and imitate. The evaluation phase measures how well each particle (candidate solution) solves the problem at hand. The comparison phase identifies the best particles. The imitation phase produces new particles based on some of the best particles previously found. These three phases are repeated until a given stopping criterion is met. The objective is to find the particle that best solves the target problem.

4 Problem Statement

Machine grouping, another name given for this kind of problems, is considered as an optimization problem. First let us introduce the elements of this optimization

model. The grouping problem was formalized by Burbidge [3] who developed a series of heuristic methods. Various approaches to cell formation have since been reported. Most of these have used the machine-component incidence matrix.

Given an incidence matrix A = $[a_{ij}]$, where:

a_{ij}= 1, if the j_{th} component visits i_{th} machine.

a_{ij} = 0, otherwise.

When a machine-component incidence matrix is constructed, no machine groups or part families are easily visible. As it was commented before, the main objective in machine grouping is the formation of set of machines and workpieces in groups so that the number of intercell transportation of pieces is minimized. Therefore, the initial matrix (Table 1) has to be transformed into a matrix that has a block diagonal structure (Table 2). This rearrangement aims at minimization of total intercell moves and minimization of within-cell load variation. A rigorous mathematical formulation of machine-component grouping problem with these objectives is given in [4].

Table 1. Initial incidence matrix

0	0	0	0	1	1	1	1
1	0	1	1	0	0	0	0
1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	0
0	0	0	0	0	1	1	1
1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1
0	1	1	1	0	0	0	0

Table 2. Rearranged matrix

1	1	1	0	0	0	0	0
1	0	1	1	0	0	0	0
0	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	0
0	0	0	0	0	1	1	1
0	0	0	0	0	1	1	1
0	0	0	0	1	1	1	1

The optimization model is stated as follows. Let M be the number of machines, P the number of parts, C the number of cells, i the index of machines (i = 1,..,M), j the index of parts (j =1,..,P), k the index of cells (k=1,..,C), $A = [a_{ij}]$ the M x P binary machine-part incidence matrix, Mmax the maximum number of machines per cell; We selected as the objective function to be minimized the number of times that a given part requires to be processed by a machine that does not belong to the cell that the part has been assigned to.

$$z_{ik} = \begin{cases} 1 \text{ if machine i} \in \text{cell k;} \\ 0 \text{ otherwise.} \end{cases}$$

$$y_{jk} = \begin{cases} 1 \text{ if part j} \in \text{cell k;} \\ 0 \text{ otherwise.} \end{cases}$$

The problem is represented by the following mathematical model:
Minimize

$$\sum_{k=1}^{C} \sum_{i=1}^{M} \sum_{j=1}^{P} a_{ij} z_{jk} (1 - y_{ik}) \tag{3}$$

Subject to

$$\sum_{k=1}^{C} (y_{ik}) = 1, \forall i, \tag{4}$$

$$\sum_{k=1}^{C} (z_{jk}) = 1, \forall j, \tag{5}$$

$$\sum_{i=1}^{M} (y_{ik}) \leqslant M_{max}, \forall k. \tag{6}$$

5 Proposed Swarm-Based Machine Grouping

A number of adaptations of traditional PSO algorithm have been implemented in order to develop a particle swarm optimization algorithm in machine grouping. First, PSO algorithm was developed for continuous domains. A discrete binary version of the PSO algorithm was developed by [14]. Correa et all. [13] proposed a Discrete PSO algorithm for attribute selection in data mining applications. The algorithm proposed here is based on the DPSO and the concept of proportional likelihoods used in [13]. The main difference between traditional PSO algorithm and the algorithm proposed here is that the proposed algorithm does not use a vector of velocities as the standard PSO algorithm does. It works with a mechanism inspired in the proportional likelihoods concept. According Correa et all. [13], the notion of proportional likelihood used in the DPSO algorithm and the notion of velocity used in the standard PSO are somewhat similar.

This algorithm deals with discrete variables (cells) and its population of candidate solutions contains particles of a given size (n=number of machines). Each component of the particle (vector) takes a value between 1 and k and represents the cell to which the machine is assigned. Potential set of solutions to the optimization problem at hand are represented by a swarm of particles. There are N particles in a swarm. X (i) keeps a record of the best position it has ever attained. This information is stored in a separated particle labeled as B (i). The swarm also keeps a record of the global best position ever attained by any particle in the swarm. This information is also stored in a separated particle labeled G.

The initial population of particles is generated by producing a series of integer random numbers. These numbers are uniformly generated between 1 and n inclusive. Potential solutions (particles) to the target problem are encoded as fixed length discrete strings; i.e., X (i) = (x(i;1); x(i;2);.....; x(i;n)), where x(i,j) $\in$ 1,....,k, i =1, 2,...,n and j = 1, 2, ..., P.

In addition a perturbation subset has been incorporated to the swarm in each iteration. This perturbation subset is inspired in the bit change mutation approach [14] with which the proposed PSO algorithm can escape from local optima. The perturbation subset corresponds to a fraction of the total number of particles of the swarm that are mutated when the fitness function tends to premature stabilization.

The perturbation subset is selected randomly from the total swarm and its size is obtained through sensibility studies as it will be shown further in this work. The perturbation subset operates as follows:

while (perturbationelement $\neq$ perturbationsetsize)

if (x(i,j)= k) then (x(i,j)= p , p $\neq$ k

end while

where p is selected randomly from the possible other cells.

For example, given the number of possible cells k=3 and N= 4, a swarm could look like this:

X (1) = (1; 2; 1; 2; 3; 3)

X (2) = (1; 1; 2; 1; 2; 3)

X (3) = (1; 2; 3; 2; 1; 3)

X (4) = (1; 2; 2; 3; 1; 3)

In this example, particle X (1) = (1; 2; 1; 2; 3;3) represents a candidate solution where machines 1 and 3 are located to cell number 1; machine 2 and 4 are located in cell 2 and machine 5 and 6 are assigned to cell number 3. After the initial population of particles is generated the process of calculation of the fitness function.

As it was commented previously, the algorithm proposed in this paper is based on the notion of proportional likelihoods and is an adaptation of the concept of velocity proposed by Correa et all.[13]. The updating process is based on X (i), B (i) and G and works as follows. In the context of combinatorial optimization the velocity of a particle must be understood as an ordered set of transformations on a solution. The transformation of a solution is represented with a term that represent the difference between two positions. Hence, in each case (X(i)-B(i)) and (X(i)-G) represent the necessary movements to change from the position given by the first term, minuend, to the position given by the second, subtrahend.

For example, consider the following instances of X(i) and B(i)

X (i) = (1; 3; 3; 2; 1; 3)

B (i) = (1; 1; 3; 3; 2; 3)

The difference between X(i) and B(i) represents the changes that will be needed to move the particle i from X to B. The number μ represents the number of elements different than 0 in the subtraction of B from X.

If the difference between a given element of X(i) and B(i) is different from 0, it means that there is a possibility of change of that position. In other words, that position is susceptible of change through the operations that are described as follows. A new vector P, is generated that register the positions of the elements equal to zero. A random number is generated and assigned to β. This number β corresponds to the number of changes that will be made to X(i) based on the difference between X(i) and B(i), therefore β is in the interval $(0,\mu)$. Then, a set ψ of β binary number is defined. If the binary number is 1 the change is made, in other hand, if the number is 0, the change is not performed. A similar process is performed to update the particle position in accordance to the best global position (G). In case that several of the applied movements involve the same position (machine), the change caused by the global best position, the second operation in our algorithm, has the priority. To illustrate, suppose that:

X(i) - B(i) = (1-1 ; 3-1;3-3; 2-3; 1-2; 3-3) = (0, 2,0,-1,-1,0)

The new vector P(i) is generated

P(i) = (2,4,5)

Thus μ= 3.

Suppose that β =2 and ψ = (0,1,1)

That means that positions 4 and 5 will be replaced in X(i) by the elements (in the same positions) of B(i). The position 2 will remain unchenged.

X‘(i) = (1 3 3 3 2 3)

The process is repeated with the new position X’(i) and the same procedure is applied with respect to G (the Global Best) obtaining the new position of X(i+1). The updating process will stop if all particles are at the same position, i.e. μ= 0.

6 Experiments and Results

The purpose of this section is to show, through a numerical example how the proposed formulation can be used in design a cellular manufacturing system. Firstly, in order to determine the parameters of the proposed algorithm, an experimental design was created. Five possible swarm sizes and five sizes of the perturbation subset was tested (table 3). The experiments consisted in running 30 times the algorithm for every swarm size and size of the perturbation subset. After running the experiments analysis of experimental results was performed. The results are shown in table 4. It can be seen that the best levels for the tested parameters are: swarm size = 1000 particles with a 5% of the size constituting the perturbation subset. These values were used for the rest of the experiments.

Secondly, a set of ten test problems were used to evaluate the proposed implementation of the PSO algorithm [5]. Each problem was solved with five different values of the maximum number of machines per cell (Mmax) giving a total of 50 problems. Table 4 shows a comparison among optimal solutions for these 10 problems obtained from [5], the solutions obtained through the use of Simulated Annealing and the solutions obtained with the proposed algorithm. As can be observed, the proposed PSO algorithm was able to find the optimal solution

Table 3. Experimental parameters for obtaining parameters for proposed algorithm

Swarm Size	Perturb. Set Size %	Average Result	Stan. Deviat.
100	10	15,8	0,5
100	20	14,9	3,44
100	5	15	2,92
100	25	16,4	6,34
200	5	16,5	3,75
300	5	15,7	3,38
400	2,5	14,7	3,33

Table 4. Comparisons among results obtained from experimental tests

Problem no.	Average Result	St. Dev.	Optimal Solution
1	11,9	1,6	11
2	9,4	1,76	7
3	7,3	1,94	4
4	15,2	1,19	14
5	11,9	1,86	9
6	7,5	1,52	5
7	8,6	1,42	7
8	15,8	2,16	13
9	11,7	2,82	8
10	10,1	1,35	8

Table 5. Comparisons among results obtained using simulated annealing, PSO and the optimal ones

Mmax=8			Mmax=9			Mmax=10			Mmax=11			Mmax=12		
opt	s.a.	p.s.o	opt	s.a.	p.s.o.	opt	s.a.	p.s.o.	opt	s.a.	p.s.o.	opt	s.a.	p.s.o.
11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
7	7	7	6	6	6	4	10	5	3	4	4	3	3	4
4	5	5	4	4	4	4	4	5	3	4	4	1	4	3
14	14	15	13	13	13	13	13	13	13	13	13	13	13	13
9	9	10	6	6	8	6	6	6	5	7	5	4	4	5
5	5	5	3	3	3	3	3	3	3	3	4	2	3	4
7	7	7	4	4	5	4	4	5	4	4	5	4	4	5
13	13	14	10	20	11	8	20	10	5	11	6	5	7	6
8	13	9	8	8	8	8	8	8	5	8	5	5	8	8
8	8	9	5	5	8	5	5	7	5	5	7	5	5	6

in 21 of the 50 solved problems. The percentage of success can be increased by solving the test problems several times. The mean error or deviation between the optimal solution and the obtained by the Simulated annealing corresponds to 1,12 (exceptional elements), while the mean error obtained from PSO algorithm

corresponds to 0,78. Also, standard deviations of the errors (in both cases) are 2,5 and 0,8 respectively. The complete set of results is shown in Table 5.

7 Conclusions

In this paper, a new approach based on Particle Swarm Optimization algorithm has been proposed to manufacturing cell formation problem. A novel discrete PSO algorithm is proposed and applied for the computation of the velocity vector as in the traditional PSO algorithm. The proposed method works with a new notion of the velocity component and incorporates the bit change mutation mechanism through the use of a subset of perturbations as possible solutions in every iteration. The PSO algorithm is hybridized with an approach called the Proportional Likelihoods which is used in data mining experiences. The algorithm and its theoretical concepts are explained and illustrated for a cell formation problem with ten instances of initial incidence matrix, 16 machines and 30 components. A set of experiments was performed that show that the algorithm is stable and presents low variability. The preliminary results obtained in this research are auspicious. In future steps the proposed algorithm will be applicable to a variety of different test problems. In addition, parameter optimization and other hybrid approaches are also topics for future research.

References

1. Selim, H.M., Askin, R.G., Vakharia, A.J.: Cell formation in group technology: review, evaluation and directions for future research. International Journal of Computers and Industrial Engineering 34(1), 3–20 (1998)
2. Lee, E.S., Pai, P.F.: Operations Research in the Design of Cell Formation in Cellular Manufacturing Systems. In: Misra, J.C. (ed.) Uncertainty and optimality: Probability, Statistics and Operations Research. World Scientific, Singapore (2002)
3. Burbidge, J.L.: Production Flow Analysis - For Planning Group. Technology, Oxford Science Publications, Oxford (1989)
4. Dimopoulos, C.: A Review of Evolutionary Multiobjective Optimization Applications in the Area of Production Research. In: Proceedings of the Congress on Evolutionary Computation (CEC 2004), Oregon, USA, June 2004, vol. 2, pp. 1487–1494. IEEE Press, Los Alamitos (2004)
5. Boctor, F.: A linear formulation of the machine-part cell formation problem. International Journal Production Research 29(2), 343–356 (1991)
6. Chen, W.-H., Srivastava, B.: Simulated annealing procedures for forming machine cells in group technology. European Journal of Operational Research 75, 100–111 (1994)
7. Venugopal, V., Narendran, T.T.: A genetic algorithm approach to the machine component grouping problem with multiple objectives. Computers and Industrial Engineering 22(4), 469–480 (1992)
8. Gupta, Y., Gupta, M., Kumar, A., Sundaram, C.: A genetic algorithm-based approach to cell composition and layout design problems. International Journal of Production Research 34(2), 447–482 (1996)

9. Aljaber, N., Baek, W., Chen, C.-L.: A tabu search approach to the cell formation problem. Computers and Industrial Engineering 32(1), 169–185 (1997)
10. Lozano, S., Adenso, B., Salinas, I., Giménez, L.: A One-Step Tabu Search Algorithm for Manufacturing Cell Design. Journal of the Operational Research Society 50(5), 509–516 (1999)
11. Andres, C., Lozano, S.: A particle swarm optimization algorithm for part-machine grouping. Robot Cim-Int. Manuf. 22(5-6), 468–474 (2006)
12. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceeding of the IEEE International Conference on Neural Networks, Perth, Australia, pp. 12–13. IEEE Service Center (1995)
13. Correa, E.S., Freitas, A., Johnson, C.G.: A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set. In: M.K., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference GECCO 2006, Seattle, WA, USA, July 2006, ACM Press, New York (2006)
14. Lee, S., Park, H., Jeon, M.: Binary Particle Swarm Optimization with Bit Change Mutation. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 90(10), 2253–2256 (2007)

Estimation of Parameters in Cox's Proportional Hazard Model: Comparisons between Evolutionary Algorithms and the Newton-Raphson Approach

David González[1], Manuel Piña[2], and Luis Torres[1]

[1] COMIMSA,
Ciencia y Tecnología 790 Frac. Saltillo 400, CP 25290,
Saltillo, Coahuila, México
Ph.: (844) 411-3200 Ext.: 1217 | Fx.: (844) 416-9346
davidgonzalez@comimsa.com
[2] UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ
Henri Dunant 4016 Zona Pronaf, CP 32310,
CD. Juárez, Chihuahua, México
Ph.: (656) 688-2100 To 09

Abstract. The numerical Newton-Raphson approach used to optimize the partial likelihood function in Cox's proportional hazard model does not converge when there are collinearity problems in the design matrix, so we proposed the use of evolutionary algorithms where a comparison is made using both approaches, taking into account the capacity of the model to make good hazard predictions. The results indicate similar performance when the Newton–Raphson model has a good starter point; otherwise, the evolutionary algorithms generate suboptimal solutions.

Keywords: Cox's Proportional Hazard Method, Newton Raphson, Genetic Algorithm, Evolutionary Algorithm of Normal Distribution (Evonorm).

1 Introduction

For modeling the factors that influence the reliability of a product the Proportional Hazards Model proposed by Cox is used [1], which is a product of two components: the failure rate function $\lambda_0(t)$, that is expressed in failures per unit of time and the component $e^{\sum \beta_i x_i}$ that depends on covariates. This model is the one most commonly used to represent the effects of a set of explanatory variables on the variable time of change, which is conditioned by the likelihood of change represented by the risk function $\lambda_0(t)$ [2]. The advantage of the Cox model is that it allows us to get as much information from each change through the change sequence and the temporary time as it occurs, and also allows us to use information from observations in which there has been no change, in other words, incomplete data [3]. Due to the existence of incomplete data, the Cox model parameters can not be estimated by the regular method of maximum likelihood because the specific form of the arbitrary risk function is

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 513–523, 2008.

unknown. For this case, Cox [1] proposed an estimation method called partial likelihood, with conditional and marginal likelihood special cases of the above. The partial likelihood method differs from the regular maximum likelihood in the sense that while the regular method is based on the product of the likelihoods for all individuals of the sample, and the partial likelihood method is based on the product of the likelihoods for all changes occurred. To estimate coefficients β_j in the Cox model, in the absence of knowledge of $\lambda_0(t)$, he proposed a partial likelihood $L(\beta_j)$, which is not a true likelihood function because it can not arise as the probability of any result observed under study [1], it can be treated as a regular likelihood function for the purpose of estimating β_j [4]. Since the Cox model is used when the magnitude and effects of concomitant variables are of interest, taking control of the temporary direction [3], the parameters β_j of the partial likelihood function are of paramount importance for determining risk. For making the estimation of these parameters it is necessary to maximize the partial likelihood function $LnL(\beta_j)$, which is done through a numerical method because of the complexity of such functions, which represents a weakness of the model due to its total dependence on the iterative method, namely if the method of solution is not efficient, results and conclusions arising out of the parameters obtained will be inefficient. Generally, optimizing the function $LnL(\beta_j)$ is done by means of the Newton Raphson numerical method, for example NCSS 2007 software uses this method to solve the function. On the other hand, you have prior knowledge that in some cases Newton's method presents divergence problems [5], therefore the parameters obtained in some cases are wrong.

The convergence of the Newton's algorithm is guaranteed by the Newton-Kantorovich theorem [6], if certain requirements are met. Of these, the most important is that the starting point must be sufficiently close to the solution and within a certain compound, where F(x) is continuously distinguishable. This algorithm has the advantage of rapid convergence if the initial conditions x(0), are close to the solution being sought. That is why a bad choice of initial points can give certain convergence problems, to name a few, a) the method never converges, b) the method converges towards a false solution, c) the method converges after many iterations (converges slowly). In an attempt to improve the convergence of Newton's method, literature speaks of modified methods of Newton, some of them based on criteria optimization. These methods still assume an improvement over the Newton's method while not ensuring global convergence to the correct solution because they can converge towards local maximum [6].

Regarding the problem explained, Thomas [7], found that existing approximation models similar to the Cox method, specifically the Breslow and Efron case, found biased estimates, for this reason they suggested a method called "MS-algorithm" in which an equation named "score equation" is proposed that is used to obtain parameters β_j, where the solution of that equation is performed using Newton's numerical method. The article made a comparison between existing methods and the proposed method, emphasizing the importance of data analysis through repetition. The simulation conducted to compare the methods showed that the approximation of Efron is the best choice for failure analysis through repetitions, mentioning also that the Breslow approximation works better if the covariant follows a uniform or binary distribution. Moreover, the proposed algorithm "MS-Algorithm" in all cases leads to results similar to the Efron method , making it a good choice for analysis with these characteristics.

Judy [8] says that some data are taken from populations in which some subjects are susceptible to events of interest and others not. This model uses maximum likelihood techniques to estimate the regression parameters and the impact on a mixed model $L_c(b, \beta, \Lambda_0; y)$, using a non-parametric likelihood and the EM algorithm. The Newton Raphson method for obtaining the parameters simultaneously was used, and proved to be very sensitive to different initial values, causing divergence in the method. The author mentions that the most efficient method that was found was the two- step Newton Raphson proposed by Prentice and Gloecker in 1978, where values for parameters α and β are obtained alternately.

On the other hand, we found a work for the optimization of a function, although it is true, and is not exactly the Cox model, the optimization method is interesting and could be applied to the Cox model. In his work, Sevcovic [9] proposed a method called Minmax which has 2 phases, for the estimation of the parameters of a model of an interest rate CIR (Cox, Ingresoll and Ross). For the optimization they proposed a function of cost U and earned a minimum global. By combining the information obtained to minimize the functional cost, U, they introduced the concept of a restricted likelihood function.

In this approach, this was done by optimizing the logarithm of a function formed by three parameters for which the function cost of U (transformed into 3 variables) reached the lowest value overall. Then they found an overall ceiling of the likelihood function proposed, which obtained optimal settings for the parameters of CIR. With regard to the optimization method, the author mentions that there are standard methods or optimization techniques for obtaining overall minimums, such as the Newton-Kantorovich gradient method, however these methods converge toward a local minimum. In addition to finding an overall minimum, their convergence is very slow. The optimization method used was based on a variant of an evolutionary algorithm strategy.

At each step of the optimization, approximating the overall minimum obtained by the evolutionary algorithm is improved by a correction that involves the application of the Newton-Kantorovich gradient method. This method is applied after generating the vector selection of the new generation, in which each intermediate parent is improved through *m* steps of the Newton-Kantorovich gradient method, obtaining a new improved vector for parents.

The results of experiments performed in this study show that in cases where the data matrix is poorly conditioned and therefore presents a collinearity problem, Newton's method converges toward a local maximum or a false solution causing inefficient conclusions at the risk obtained by means of the Cox model.

2 Cox's Proportional Hazard Model

The proportional hazard model proposed by Cox [1] for modeling the relation between the failure time and the covariant variables Z, is given by:

$$\lambda(t / Z) = \lambda_0(t) \exp\left(\beta^T Z\right), \tag{1}$$

Where $\lambda_0(t)$ is a function of the unknown hazard that correspond to Z = (0,...,0), and $\beta = (\beta_1,..., \beta_p)^T$ is the estimated vector of regression coefficients. The interpretation of

the β_j parameters in this model is similar to the interpretation of another regression model. For the estimation of these regression parameters, when the failure times are censored data, Cox, in 1972, proposed the partial likelihood method defined in (2):

$$\prod_{j=1}^{N} \frac{\exp\left(\beta^T Z_{(j)}\right)}{\sum_{i \in R_j} \exp\left(\beta^T Z_i\right)} \tag{2}$$

For a given function (2), the assumptions are that there are no ties between failure times, $t_1<...<t_N$, where N denotes the order of the observed times and (j) denotes the case of a component which fails at t_j. So if R_j represents the risk set at time t_j with $R_j = \{i : X_i \geq t_j\}$. The partial log likelihood function for (2) is given by:

$$L(\boldsymbol{\beta}) = \sum_{j=1}^{N} \left\{ \boldsymbol{\beta}^T Z_{(j)} - \log\left[\sum_{i \in \Re_j} \exp(\boldsymbol{\beta}^T Z_i) \right] \right\} \tag{3}$$

Likewise, if the system presents N distinct failure times, $t_1<...<t_N$, and each time t_j ($1 \leq j \leq N$) there exist d_j observed failures, where D_j represents the set of all individuals who failed at the same time t_j and R_j represents the risk set at time t_j, with $R_j = \{i : X_i \geq t_j\}$. And as a result of these repetitions, the estimation of the maximum likelihood function given in (3) becomes complicated or not feasible. There are two common widely accepted approaches of the likelihood function given in (3). One of them is Breslow's partial likelihood functions given by [10]:

$$L_B(\boldsymbol{\beta}) = \sum_{j=1}^{N} \left\{ \boldsymbol{\beta}^T \sum_{I \in D_j} Z_I - d_j \log\left[\sum_{i \in \Re_j} \exp(\boldsymbol{\beta}^T Z_i) \right] \right\} \tag{4}$$

For the function defined in (4), the literature states that this approximation works well when there are not many ties. When there are many ties, it is suggested to use Efron's likelihood function given by:

$$L_B(\boldsymbol{\beta}) = \sum_{j=1}^{N} \left\{ \boldsymbol{\beta}^T \sum_{I \in D_j} Z_I - \sum_{k=1}^{d_j} \log\left[\sum_{i \in \Re_j} \exp(\boldsymbol{\beta}^T Z_i) - (k-1)/d_j \sum_{i \in D_i} \exp(\boldsymbol{\beta}^T Z_i) \right] \right\} \tag{5}$$

Indeed, the function given in (4) is easy to use and is therefore more popular, however, Efron's approximation given in (5), is generally the more accurate. Also, both likelihoods reduce to the partial likelihood given in (3), when there is no tie.

2.1 Estimating the Hazard and Survival Functions

In reliability, the cumulative baseline hazard function that represents the probability that an analyzed system fails at time t_i, could be estimated by [10]:

$$\hat{\Lambda}_0 = \sum_{j:t_j \leq t} \frac{\delta_j}{\sum_{i \in \Re_j} \exp(\boldsymbol{\beta}^T Z_j)} \tag{6}$$

where, $\delta_j = I\ (T_j \leq C_J)$. The function $\hat{\Lambda}_0(t)$ is a right-continuous step function with jumps at the observed failure times, and it is often referred to as the Breslow estimator. In the case of tied events, each of the subjects in a tie contributes its own term to the sum, and this term is the same for all subjects who failed at the specific time. This estimator can also be derived through a profile likelihood approach [4] and the baseline survival function $S_0(t) = \exp[-\Lambda_0(t)]$ can thus be estimated by $\hat{S}_0(t) = \exp[-\hat{\Lambda}_0(t)]$. The estimated survival function of an individual with covariate value Z_j is given by:

$$\hat{S}_0(t/z) = \exp\left[-\hat{\mathbf{\Lambda}}_0(t)e^{\hat{\boldsymbol{\beta}}^T Z}\right] \tag{7}$$

Unfortunately, the efficiency of the risk estimation given in (6) depends on the efficiency of the numerical method deemed regression coefficients. Usually this estimate is done through the Newton method.

3 Newton's Method

The Newton method was described by Isaac Newton in *De analysi per aequationes numero terminorum infinitas*, written in 1669 and published in 1711 by William Jones and by *De metodis infinitarum*, written in 1671, translated and published as a fluxions method in 1736 by John Colson. The method is *fluxionum et serierum* obtained through the Taylor expansion series for n variables around a root with respect to $\hat{x} = \hat{x}_i$, as follows [5]:

$$f(\hat{x}) = f(\hat{x}_i) + (\hat{x} - \hat{x}_i)'\left[\frac{\partial f(x)}{\partial x}\Big|_{x=\hat{x}_i}\right] + \frac{1}{2}(\hat{x} - \hat{x}_i)'\left[\frac{\partial^2 f(x)}{\partial x \otimes \partial x'}\Big|_{x=\hat{x}_i}\right](\hat{x} - \hat{x}_i) + o\left(\left\|\hat{x} - \hat{x}_i\right\|^2\right) \tag{8}$$

This expansion can be written as:

$$f(\hat{x}) \approx f(\hat{x}_i) + (\hat{x} - \hat{x}_i)' g_{\hat{x}_i} + \frac{1}{2}(\hat{x} - \hat{x}_i)' H_{\hat{x}_i}(\hat{x} - \hat{x}_i), \tag{9}$$

Where,

$$g_{\hat{x}_i} = \left[\frac{\partial f(x)}{\partial x}\Big|_{x=\hat{x}_i}\right] \quad \text{y} \quad H_{\hat{x}_i} = \left[\frac{\partial^2 f(x)}{\partial x \otimes \partial x'}\Big|_{x=\hat{x}_i}\right] \tag{10}$$

To find the vector that optimizes function $f(x)$, set the derivative of $f(\hat{x}) = 0$ and solve for $\hat{x}$. The result is:

$$\frac{\partial f(\hat{x})}{\partial \hat{x}} = 0 \Rightarrow g_{\hat{x}_i} + H_{\hat{x}_i}(\hat{x} - \hat{x}_i) = 0 \tag{11}$$

Solving for $\hat{x}$ given:

$$\hat{x} = \hat{x}_i - H_{\hat{x}_i}^{-1} g_{\hat{x}_i} \tag{12}$$

In practice, this formula given by (12) allows us to define an iterative process for functions of several variables.

$$x^{k+1} = x^k - H^{-1}(x^k)g(x^k) \tag{13}$$

The Newton algorithm usually converges to the optimal solution, however, its disadvantage is that when matrix H defined in (10) is poorly conditioned; the solution differs because in its early iterations the increases are large. The convergence of the algorithm is insured by the Newton-Kantorovich theorem [6] if certain requirements are fulfilled, of which the most important is that the initial point should be sufficiently closer to a solution within a certain ground, where F(x) is continuously differentiable. That is why when a process operation (initial point), is far from optimal its analysis will have divergence problems. In general the following cases could be submitted:

- The method is divergent
- The method converges towards a false solution
- The method converges after many iterations.

In order to conduct a thorough analysis of the trajectory of iterations, from the initial point to the peak, algorithms have been developed whose application presents efficient results for this aim, including evolutionary algorithms.

4 Evolutionary Algorithm of Normal Distribution Function (Evonorm)

Evonorm is an easy implementation of an estimation of distribution algorithm (EDA) because it uses a normal distribution function per parameter where the mean and standard deviation of this normal distribution are calculated from a set of parameters selected and previously generated by the same normal distribution function [11]. Evonorm has one principal difference with other EDAs, because Evonorm uses the best solution found with a probability of 50% and for the other 50% Evonorm uses the mean of the best solutions. In this version of Evonorm, independence between variables is assumed.

5 Genetic Algorithm (GA)

Genetic algorithms (GA) are the most used evolutionary algorithms. GA was developed by John Holland [14], 1975, and has grown to be the most used paradigm to solve optimization problems (i.e. Mitsuo Gen and Runwei Chen, 2000). There are several variants of the GA; nevertheless, all have four general procedures: evaluation of the individuals, selection of the best individuals (in a deterministic or stochastic way), crossover, and mutation of individuals [15]. Every individual is a solution represented as a binary vector and a set of solutions represents a population of potential solutions or individuals in an analogy to natural processes. The simple GA has binary vectors to represent parameters so real parameter representation requires a decoding procedure for it to be used in the evaluation procedure. The population begins with random solutions usually with low performance and high diversity. The evolutionary procedures (evaluation, selection, crossover and mutation) are applied cyclically way. Solutions of better performance are obtained and the diversity is lost. It is a more intelligent way to search for solutions than the “trial and error” procedure.

6 Experiment

The experiment consisted in the estimating of the Cox model parameters for two cases in particular, using for optimizing the partial likelihood function given in (4): Newton's method, a Genetic Algorithm and an evolutionary algorithm Evonorm, in order to compare the results obtained. In the case of the genetic algorithm, a binary codification is used with 16 bits per parameter. A population of 100 individuals is used where every chromosome has a size of 48 bits. We considered a one point crossover (with a probability of 0.9) and a single mutation (with a probability of 0.052) setup by experience. A tournament selection of size two is used in the algorithm that runs in 100 generations.

Table 1. Failure time sample for case 1

M. Tool	Speed	Feed	Depth	T	M. Tool	Speed	Feed	Depth	T
1	340	0.0063	0.0210	0.0210	9	440	0.0091	0.0290	0.0290
2	570	0.0063	0.0210	0.0210	10	440	0.0091	0.0290	0.0290
3	340	0.0142	0.0210	0.0210	11	440	0.0091	0.0290	0.0290
4	570	0.0142	0.0210	0.0210	12	440	0.0091	0.0290	0.0290
5	340	0.0063	0.0400	0.0400	13	305	0.0091	0.0290	0.0290
6	570	0.0063	0.0400	0.0400	14	635	0.0091	0.0290	0.0290
7	340	0.0142	0.0400	0.0400	15	440	0.0047	0.0290	0.0290
8	570	0.0142	0.0400	0.0400	16	440	0.0173	0.0290	0.0290

Case 1: We used the design data published by Taraman in 1974, and then analyzed by Merrick, Soyer and Mazzuchi in 2003, where T is the tool life in minutes and X_1, X_2, X_3 are speed (speed fpm), feed rate (feed ipr) and depth of cut (depth in inches). The data is shown in Table 1.

Table 2 shows that the three optimization methods found similar results, which is due to the stability of the data used as influencing factors contained in the matrix X_i, so the inferences to be made about tool reliability as well as the risk using parameters that maximize the function, are indistinct for all three methods.

Table 2. Comparison for both cases

Method	Case 1				Case 2			
	Lk	**B_1**	**β_2**	**β_3**	**Lk**	**β_1**	**β_2**	**β_3**
Newton	-35.6000	0.0256	209.1475	76.3692	-0.4553	0.2751	-1.3455	-2.2465
GA	-35.6000	0.0256	209.0117	76.2023	-0.000000398	1.9572	-9.8328	4.4133
Evonorm	-35.6013	0.0257	207.0278	75.9488	-0.000000309	2.0000	-10.0000	2.2802

Case 2: On the other hand, for obtaining the partial likelihood function of the Cox model we used the data presented in Table 3, which have collinearity problems between its variables. As a measure of collinearity, there are some diagnostic methods, such as the eigensystem analysis which includes the number of condition k. This

value must be less than 1,000 and the variance inflation factors (VIF) must be less than 10 [17]. For data pertaining to case 2, the number of condition k=79362000 and inflation factors are for X_1, VIF=12,076, X_2 has a VIF=1,062 and X_3 has a VIF=12,076 indicating that the data presented in Table 3 has a severe problem of collinearity.

Table 3. Sample for case 2

No.	X_1	X_2	X_3	T	No.	X_1	X_2	X_3	T
1	1300	7.50	0.012	49.00	9	1200	11.00	0.032	34.50
2	1300	9.00	0.012	50.20	10	1200	13.50	0.026	35.00
3	1300	11.00	0.012	50.50	11	1200	17.00	0.034	38.00
4	1300	13.50	0.013	48.50	12	1200	23.00	0.041	38.50
5	1300	17.00	0.014	47.50	13	1100	5.30	0.084	15.00
6	1300	23.00	0.012	44.50	14	1100	7.50	0.098	17.00
7	1200	5.30	0.040	28.00	15	1100	11.00	0.092	20.50
8	1200	7.50	0.038	31.50	16	1100	17.00	0.086	29.50

Similarly, for optimizing the partial likelihood function, the 3 methods were applied to compare their results. In this case, unlike case 1, optimal settings were different. Analyzing the values for Lk in the table 2, we note that the Newton method obtained a value of LK =-0.4553 at 560 iterations, unlike the evolutionary algorithms that from the second or fourth iteration reached better optimal settings, arriving at end values significantly above the traditional method.

In Cox's proportional hazards method, parameters interpretation is performed through the value of $e^{\beta i}$, for each β_j. This means that for each unitary increment, the risk rate increases or decreases. In addition, $100*(e^{\beta i}-1)$ provides the change percentage that occurs with each unit change in the independent variable reflected in the risk rate [2]. Table 4 shows the estimated values for each variable.

Table 4. Parameter effects

Method	**Exp(B)**			**100*(e^b-1)**		
Newton	1.316662335	0.260409471	0.105768768	31.67%	-73.96%	-89.42%
GA	7.079476753	0.00005366	82.54140118	607.95%	-99.99%	8154.14%
Evonorm	7.389056099	0.00004540	9.778694613	638.91%	-100.00%	877.87%

The estimate of exp(β) in the case of GA for β1 = 7.07, indicates that for each unitary increment in the explanatory variable X_1, the risk rate is multiplied by 7.07. In addition, 100*(eb-1) = 607.95% provides the percentage of change in the risk rate that comes with each unit change in the independent variable. In other words, for each unit that increases the risk rate also increases in a 607.95% compared to its previous value. Similarly, the interpretation is made for each of the β_j coefficients obtained in each optimization method. The most significant increment in risk was shown in β_3 parameter, which if deemed by the genetic algorithm represents an increase in the risk rate of 8154.14% per each unitary increment in the variable X_1.

If we refer to the best value of Lk for choosing the best optimization method, the best method would be Evonorm, which indicates that any unitary increment in one of its covariates generates a great increase in risk. Moreover, the results achieved through Newton's method suggest much smaller increases in the risk to any unitary increment in their covariates, which results in carrying out wrong conclusions.

The most significant difference between the conventional method and the alternative methods was specifically shown in parameter β_3 due to the change of sign. That is, while the parameter obtained through alternative methods indicates an *increase* in the risk rate, Newton's method suggests that the change percentage in the risk rate *decreases* with each unit change in the independent variable in β_3 = -89.42%. That is to say, for each unitary increment, the risk rate *decreases* by 89,420% compared to its previous value, whereas for Evonorm the rate risk increases by 877.87% compared to its previous value. To show this, calculations were done by means of (6) the corresponding risk for each method. The results are shown in Table 5.

Table 5. Risk estimation for each method

T_i	Newton		GA		Evonorm		T_i	Newton		GA		Evonorm	
	$S_0(t)$	$\Lambda_0(t)$	$S_0(t)$	$\Lambda_0(t)$	$S_0(t)$	$\Lambda_0(t)$		$S_0(t)$	$\Lambda_0(t)$	$S_0(t)$	$\Lambda_0(t)$	$S_0(t)$	$\Lambda_0(t)$
15.0	1	0	1	0	1	0	38.0	0	14.99	0	Inf	0	Inf
17.0	1	0	1	0	1	0	38.5	0	606.7	0	Inf	0	Inf
20.5	1	0	1	0	1	0	44.5	0	Inf	0	Inf	0	Inf
28.0	1	0	1	0	1	0	47.5	0	Inf	0	Inf	0	Inf
29.5	1	0	0	Inf	0	Inf	48.5	0	Inf	0	Inf	0	Inf
31.5	0.999	0.002	0	Inf	0	Inf	49.0	0	Inf	0	Inf	0	Inf
34.5	0.993	0.006	0	Inf	0	Inf	50.2	0	Inf	0	Inf	0	Inf
35.0	0.824	0.193	0	Inf	0	Inf	50.5	0	Inf	0	Inf	0	Inf

7 Conclusions

A comparison was made between three different optimization methods for maximizing the function obtained by applying Cox's proportional hazards method to prove that conventional optimization methods used in obtaining the parameters of that function can present instability, bringing as a consequence inefficient conclusions of estimated risk.

With respect to cases presented, case 1 shows estimates of the parameters that are almost identical, which have as a consequence that the results in the estimated risk for each case and inferences to be made about this will be the same for each method, due to the stability of the estimate. This shows that evolutionary algorithms are an option for optimizing the partial likelihood function in the Cox method.

The experimental results show that in case 2, where there is a collinearity problem, Newton's method converges toward a local minimum or a suboptimal solution, presenting slow convergence towards the maximum. On the other hand, the Genetic Algorithm and Evonorm estimated values well above the optimum obtained by the conventional method showing the main difference in parameter β_3, where the parameter estimated even changed its sign.

Due to the fact that β_j are vital for estimating risk in Cox's model, changing signs on parameter β_3 greatly influences the conclusions and interpretations made about the risk, that is to say, while the parameter obtained by means of evolutionary algorithms indicate a growth in the risk rate, Newton's method suggests that the change in the risk rate decreases with each unitary change in the independent variable. The results obtained through Newton's method suggest minor increases in the risk, which results in carrying out the wrong conclusions. If we consider these results in maintenance plans, planned replacements or preventive maintenance may be performed less often, causing safety risks and increased costs.

The foregoing demonstrates that evolutionary algorithms, in this case the Genetic Algorithm and Evonorm, are useful tools in optimizing the partial likelihood function used in Cox's proportional hazards method, because in case 2 better results were obtained than through the conventional method, however, it is necessary to validate these results in the field.

References

1. Cox, D.R.: Regression models and life tables. Journal of the Royal Statistical Society, Serie B (Methodological) 34(2), 187–220 (1972)
2. Harrell, F.: Regression modeling strategies: with applications to linear models, logistic regression and survival analysis. Springer Series in Statistics, pp. 389–522 (2001), ISBN 0387952322
3. Palmer, L.: Modelos de regresión de Cox: Ejemplo numérico del proceso de estimación de parámetros. Psicothema 5(2), 387–402 (1993)
4. Quigley, J.: Proportional Hazards Regression. Statistics for biology and health, pp. 129–199. Springer, Heidelberg (2008)
5. Sun, W., Yuan, Y.: Optimization theory and methods: nonlinear programming. Springer optimization and its applications 1, 119–301 (2006)
6. Nazareth, J., Borwein, J., Borwein, P.: Differentiable Optimization and Equation Solving, A Treatise on Algorithmic Science and the Karmarkar Revolution. Springer, Heidelberg (2003)
7. Scheike, T., Sun, Y.: Maximum likelihood estimation for tied survival data under Cox regression model via EM-algorithm. Lifetime Data Anal. 13, 399–420 (2007)
8. Sy, J., Taylor, J.: Estimation in a Cox Proportional Hazards Cure Model. Biometrics 56, 227–236 (2000)
9. Sevcovic, D., Csajkova, A.: On a two-phase minmax method for parameter estimation of the Cox, Ingersoll, and Ross interest rate model. CEJOR 13, 169–188 (2005)
10. Pham, H.: Springer Handbook of Engineering Statistics, ch. 21. Springer, London (2006)
11. Torres, L.: EvoNorm, A New Evolutionary Algorithm to Continuous Optimization. In: Workshop on Optimization by Building and Using Probabilistic Models (OBUPM 2006). Genetic and Evolutionary Computation Conference (GECCO 2006) CD Proceeding Tutorials and Workshops, Seattle, July 8-12 (2006)
12. Torres, L.: Evonorm: Easy and effective implementation of estimation of distribution algorithms. In: Gelbuckh, A., Guerra, S.S. (eds.) RCS, vol. 23, pp. 75–83 (November 2006)
13. Mitsuo, G., Runwei, C.: Genetic Algorithms & Engineering Optimization. John Wiley & sons Press, Chichester (2000)

14. Holland, J.: Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, Ann Arbor (1975)
15. Goldberg, D.: Genetic algorithms in search, optimization, and machine learning. Oxford University Press, New York (1989)
16. Fulcher, J., Jain, L.: Computacional intelligence. Studies in computacional intelligence, vol. 115, pp. 3–78. Springer, Heidelberg (2008)
17. Montgomery, D.C., Peck, E.A., Vining G.y.: Introducción al Análisis de Regresión Lineal. México, Editorial Continental tercera edición (2002), ISBN:970-24-0327-8

Gravitational Fuzzy Clustering

Umut Orhan[1], Mahmut Hekim[1], and Turgay Ibrikci[2]

[1] Electronics and Computer Department, Gaziosmanpasa University 60250, Tokat, Turkiye
umutorhan@mail.gop.edu.tr, mhekim@gop.edu.tr
[2] Electrical and Electronics Engineering, Cukurova University 01330, Adana, Turkiye
ibrikci@cukurova.edu.tr

Abstract. Data clustering is an important part of cluster analysis. Numerous clustering algorithms based on various theories have been developed, and new algorithms continue to appear in the literature. In this paper, supposing that each cluster center is a gravity center and each data point has a constant mass, Newton's law of gravity is transformed from m/d^2 to $1/d^2$. According to adapted the law, we have proposed novel method called Gravitational Fuzzy clustering. The three main contributions of new algorithm can be summarized as: *1*) it becomes more sophisticated technique by taking advantages of K-means, fuzzy C-means and subtractive clustering methods, *2*) it removes the dependence on initial condition by taking account of the gravitation effect, *3*) it improves the cluster centers by means of the gravity center of clusters. We illustrate the advantage of the resulting of gravitational approach with several examples.

Keywords: gravitational fuzzy clustering; k-means; subtractive clustering; fuzzy c-means.

1 Introduction

Data clustering is concerned with the partitioning of a dataset into several groups such that the similarity within a group is larger than that among other groups. This implies that the dataset to be partitioned has to have an inherent grouping to some extent; otherwise if the data is uniformly distributed, determination of data clusters will be unsuccessful, or will lead to artificial partitions. Another problem that may arise is the overlapping of data groups. Overlapping groupings sometimes reduce the efficiency of the clustering method, and this reduction is proportional to the amount of overlap between groupings. The techniques presented in literature are usually used in conjunction with other sophisticated neural or fuzzy models. In particular, most of these techniques can be used as pre-processors to determine the initial locations for radial basis functions or fuzzy knowledge rules. The common approach of all the clustering techniques is to find cluster centers that will stand for each cluster. Researchers have proposed many solutions for this issue based on different theories, and many surveys focused on special types of clustering algorithm have been presented [1], [2], [3].

The best-known and earliest partitional clustering is the K-means (KM) clustering algorithm. Its main advantageous is the capacity of fast converging. In spite of having many successful applications in several fields, it has many drawbacks. The membership

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 524–531, 2008.

values being only 0 or 1 may not always reflect the practical relationship between the data point and the cluster. In order to cope with this drawback, Fuzzy C-means (FCM) method employs fuzzy partitioning such that each data point can belong to several clusters with membership values between 0 and 1. Both clustering techniques try to partition the data into given the number of clusters. Another method, Subtractive clustering, does not need being known the number of clusters. First, the algorithm finds the largest cluster by using the density function, then the second one, and so on. There have also been many researches to improve partitional clustering such as density estimation based non-parametric clustering algorithms, the mean shift procedure to deal with the blurring process, for discrete data the convergence of the mean shift algorithm to the nearest stationary point of the underlying density function, and the variable-bandwidth mean shift [4], [5], [6], [7], [8], [9].

Most partitional clustering algorithms include predefined parameters such as the weighting exponent m and the number of clusters C in FCM clustering algorithm. Choosing the best values of them is very important, but there are only a few theoretical researches in the literature [10], [11], [12].

In this paper, a new algorithm has been developed for determining the optimum cluster centers, and it is applied a fuzzification which is independent of a weighting parameter used in common FCM. This new algorithm is based on Isaac Newton's law of gravitation. According to the law, each cluster center is a gravity center and every object attracts every other object by a force along the line intersecting both objects. The gravitation force between two objects is proportional to the masses and inversely proportional to the square of the distance between them.

We focus on an effective approach called Gravitational Fuzzy Clustering for detection of the optimum cluster centers on the basis of gravitational centers. It is useful for obtaining the initial matrices of more complex clustering methods.

The paper is organized as follows. In Section 2, the structure of the new clustering is presented. Section 3 shows results for some artificial and real datasets. Our results are discussed in Section 4. Finally, the conclusions are given in Section 5.

2 Gravitational Fuzzy (GF) Clustering

Gravitational Fuzzy (GF) clustering is a new approach to find the optimum cluster centers and to reduce computation time. It takes advantages of mentioned above three methods and takes account of the gravitation law. According to the gravitation law, the attraction between two objects is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. By inspired of this, we have supposed that each cluster center is a mass gravity center and each data point has a unit mass. The law equation m/d^2 is transformed to $1/d^2$ and the weight of ith data point is calculated by using Equation 1.

$$w_i = \sum_{j=1}^{n} \frac{1}{\left\| x_i - x_j \right\|^2} \tag{1}$$

where n is the size of dataset and $\left\| x_i - x_j \right\|$ is the distance between x_i and x_j.

If a data point has many neighbor data points, it is heavier than the others. Let D_k be refined density of kth cluster. The heaviest data point is selected as the cluster center. Next, the refined density measure of each data point x_k is revised by following:

$$D_k = \begin{cases} \max_{i=1}^{n} (w_i), & \text{if } k = 1 \\ \max_{i=1}^{n} \left(1 - \sum_{j=1}^{k-1} \frac{w_j}{\left\| x_i - CC_j \right\|^2} \right), & \text{otherwise} \end{cases} \tag{2}$$

The data points around the first cluster center CC_1 reduce refined density measurement. After revising the refined density function, the heaviest data point is selected as next cluster center. The partitioned dataset are defined by a $c \times n$ binary membership matrix U according to Equation 3:

$$u_{ij} = \begin{cases} 1 & \text{if } \left\| x_i - CC_j \right\| = \min_{k=1}^{C} \left\| x_i - CC_k \right\| \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Suppose that a dataset consists of C clusters and G_i is ith cluster, then the cost function is below:

$$J = \sum_{i=1}^{C} \left(\sum_{k, x_k \in G_i} u_{ij} \left\| x_k - CC_i \right\|^2 \right) \tag{4}$$

If CC_i is the closest center to x_k, it belongs to cluster G_i. In order to find the optimum cluster centers, new gravity approach detects the initial cluster centers and then slides them by Equation 5.

$$CC_i = \frac{1}{\left| G_i \right|} \sum_{k, x_k \in G_i} x_k \tag{5}$$

where $\left| G_i \right|$ is the size of G_i, or $\left| G_i \right| = \sum_{j=1}^{n} u_{ij}$.

Each data point belongs to all cluster centers with the ratio of distances from each cluster centers. A new membership matrix U is calculated by using new fuzzification function below.

$$u_{ij} = \frac{\lambda}{\left\| c_i - x_j \right\|^2}, \quad \lambda = \frac{1}{\sum_{i=1}^{k} \frac{1}{\left\| c_i - x_j \right\|^2}} \tag{6}$$

where λ is normalization coefficient and k is the number of data points belonging to investigated cluster G_i.

This new approach consists of following steps:

*Step*1. Calculate the weight values of each data point according to Equation (1) and select the heaviest data point as first cluster center.

Step2. Compute the reduced density values of each data point according to Equation (2).
*Step*3. Select the data point with new highest refined density as next cluster centers and go to Step2. This step is repeated *C* times after go to Step4.
Step4. Compute the binary membership matrix *U* according to Equation (3).
Step5. Calculate the cost function *J* according to Equation (4). If it is smaller than selected threshold value then go to Step7.
*Step*6. Slide the cluster centers according to Equation (5) and go to Step4.
*Step*7. Determine the fuzzy membership matrix *U* according to Equation (6).

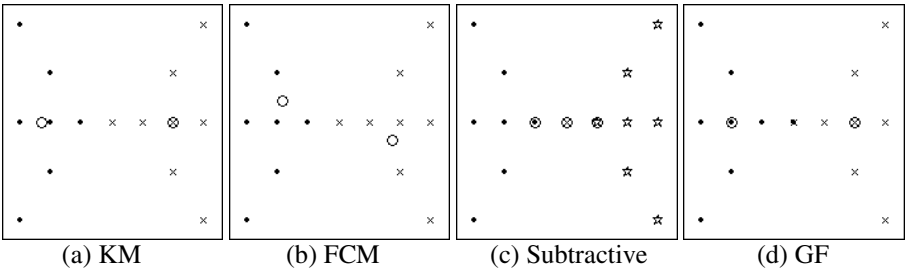
3 Numeric Examples

In this section, it has been showed several examples to illustrate the ideas presented in the previous section. It has been presented three artificial dataset examples to provide several viewpoints into this new approach. Also, in order to examine the real performance of the proposed novel method, two known multidimensional clustering benchmark problems are presented for real dataset.

3.1 Artificial Dataset

In this section, the novel GF algorithm is illustrated with several examples and applied three simple examples to provide insights into new approach. The performance of new method is compared with those of the corresponding KM, FCM and subtractive clustering methods.

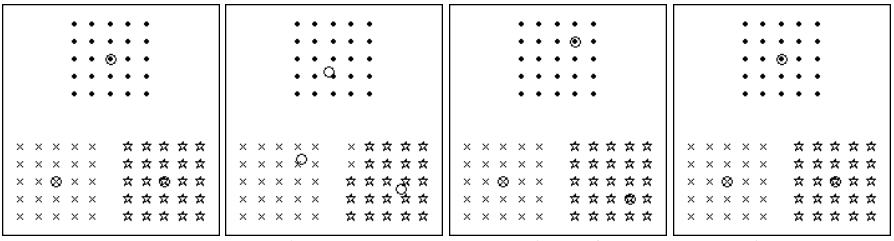
Example 1. The butterfly dataset consists of 15 two dimensional vectors with two clusters [13], which has been used as test data in many experiments and it has a symmetric data distribution. The symmetric point is concentrated on which belongs to both clusters with a membership value of (0.5). Figure 1 shows KM, FCM, Subtractive and GF clustering methods for butterfly dataset.



(a) KM (b) FCM (c) Subtractive (d) GF

Fig. 1. Results of clustering methods for Example 1

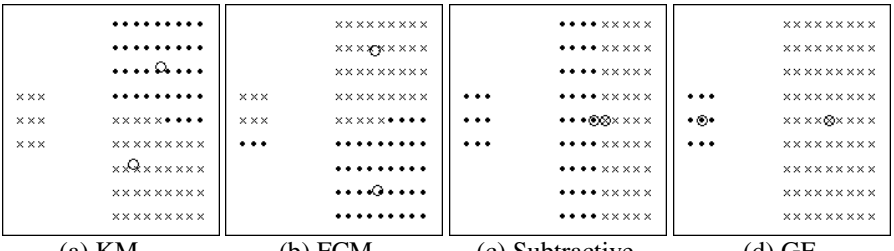
Example 2. 75 two-dimensional vectors are prepared with three clusters for another example. Figure 2 displays this artificial dataset which consists of three identical square lattices each containing 25 data points. Figure 2 shows KM, FCM, Subtractive and GF clustering methods for this dataset.



(a) KM (b) FCM (c) Subtractive (d) GF

Fig. 2. Results of clustering methods for Example 1

Example 3. 90 two dimensional vectors are prepared with two clusters for another example. Figure 3 displays this artificial dataset which consists of two non-identical lattices. One of them contains 9 data point and other 81. Figure 3 shows KM, FCM, Subtractive and GF clustering methods for this two dimensional dataset



(a) KM (b) FCM (c) Subtractive (d) GF

Fig. 3. Results of clustering methods for Example 3

3.2 Real Datasets

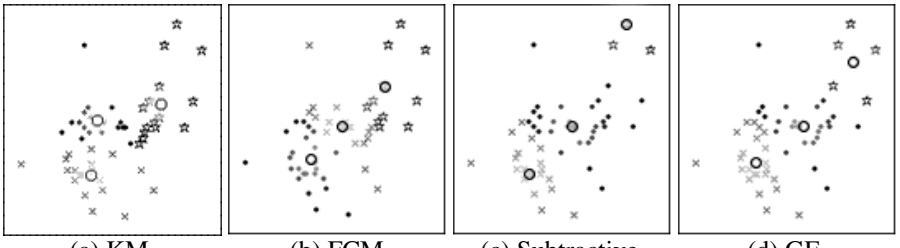
In this study, we used two different datasets, Anderson Iris Dataset and Blood Pressure Dataset. In order to study the performance of the clustering algorithms presented in the previous Section, we applied our algorithm to a benchmark.

Example 4. Anderson Iris Dataset consists of 150 five-dimensional vectors [14]. This dataset has often been used as a standard for testing clustering algorithms. The components of a vector are the measurement of the petal length and width, and sepal length and width of a particular iris plant and last fifth vector is added to describe the plant types. There are 50 plants in each of three classes of iris data and its cluster centers are in Table 1.

Table 1. The cluster centers for Anderson Iris dataset.

	KM				FCM				Subtractive				GF			
CC_1	0.19	0.59	0.08	0.06	0.2	0.58	0.08	0.05	0.22	0.58	0.08	0.04	0.2	0.59	0.08	0.06
CC_2	0.45	0.32	0.55	0.51	0.63	0.42	0.75	0.82	0.47	0.42	0.64	0.71	0.64	0.4	0.77	0.8
CC_3	0.64	0.41	0.77	0.80	0.48	0.37	0.58	0.53	0.39	0.37	0.54	0.5	0.45	0.32	0.55	0.51

Example 5. Blood pressure dataset consists of 53 two-dimensional vectors with three clusters [14]. This dataset has been used as a standard for testing clustering algorithms. Figure 4 shows KM, FCM, Subtractive and GF clustering methods for blood pressure dataset and its cluster centers are summarized in Table 2.



(a) KM (b) FCM (c) Subtractive (d) GF

Fig. 4. Results of clustering methods for Blood Pressure dataset

Table 2. The cluster centers for Blood Pressure dataset

	CC_1		CC_2		CC_3	
KM	0.38	0.21	0.42	0.50	0.77	0.59
FCM	0.30	0.18	0.50	0.46	0.70	0.62
Subtractive	0.32	0.22	0.56	0.45	0.86	1.00
GF	0.33	0.27	0.59	0.47	0.88	0.81

4 Results and Discussion

A clustering method is usually evaluated by using the artificial and the real datasets. It must be able to solve not only various datasets, but also datasets with different feature and sampling size. Therefore, we have used many special datasets in the numeric examples. Artificial datasets used in the literature have some properties such as *symmetric, discrete*, and *identical*. In order to generalize its success, novel method is checked by two real asymmetric and non-discrete datasets after being tested by artificial datasets. One of them has two features; the other has five features.

KM and FCM clustering can not provide fixed cluster centers because of depending on initial condition. Therefore, they should be run several times for all datasets. Both Subtractive and GF clustering have only one solution independent of initial condition; consequently, it is enough to run once for them.

In this study, subtractive clustering could not determine the optimum cluster centers for first three datasets, and it miscalculates the number of clusters for first dataset because of one symmetric data point with equal membership. FCM and GF clustering find two clusters; whereas, subtractive clustering three clusters for Example 1. FCM clustering could not detect the optimum cluster centers because of dependence on initial condition mentioned above. By means of finding mathematical gravity centers, GF clustering could have calculated both the number of clusters and the memberships correctly.

The reason for preference of Anderson Iris dataset with 150 samplings and 5 features is big size for testing the methods. The results of other methods are similar to results of GF clustering for this dataset, which proves the success of GF clustering.

All clustering methods find three clusters for blood pressure dataset which is a nondiscrete-asymmetric dataset. Because subtractive clustering tries the discreteness, three points at the right top are assigned to a cluster. KM and FCM clustering tend to homogenous distribution; therefore clusters have nearly same the number of member. GF clustering takes account of both discreteness and homogenous distribution; so the result has a similar appearance to average of other three methods.

5 Conclusion

The aim of clustering is to obtain the most different groups in a dataset. A clustering method finds the similar data points and puts them into groups. If the groups in a dataset are found, dataset can be represented by fewer symbols. In the literature, researchers have proposed many solutions for this issue based on different theories. But there are still some problems such as the optimum cluster centers and the computation time.

In this paper, we have presented a new clustering method based on Newton's gravitation law. It has taken advantage of KM, FCM and Subtractive clustering methods. A different fuzzification has been also used in new method. The new method has been performed by three artificial and two real datasets. Datasets used in Examples 1, 2 and 3 are discrete and artificial; the next ones are non-discrete and real. Experimental results show that new algorithm improves the cluster centers and reduces the computation time.

Finally, the novel method is successful for not only symmetric and identical datasets, but also asymmetric and non-identical datasets. It also removes dependence on initial conditions in contrast to common clustering methods.

References

1. Hekim, M., Orhan, U.: A Validity Measure for a New Hybrid Data Clustering. In: International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, pp. 70–74 (2007)
2. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
3. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 4–36 (2000)
4. Berkhin, P.: Survey of Clustering Data Mining Techniques. Grouping Multidimensional Data, 25–71 (2006)
5. Cheng, Y.: Mean Shift, Mode Seeking, and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(8), 790–799 (1995)
6. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
7. Comaniciu, D.: An Algorithm for Data-Driven Bandwidth Selection. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(2), 281–288 (2003)

8. Tsoukalas, L., Uhrig, R.: Fuzzy and Neural Approaches in Engineering. John Wiley & Sons, Inc., NY (1997)
9. Yu, J.: General C-Means Clustering Model: A Unified Approach to the Theories and Methods of Partitional Clustering Algorithms. Technical Report, AI Lab, Northern Jiaotong Univ., Beijing (2003)
10. Baraldi, A., Blonda, P.: A Survey of Fuzzy Clustering Algorithms for Pattern Recognition-Part I and II. IEEE Transactions on Systems, Man and Cybernetics, Part B 29(6), 778–801 (1999)
11. Baraldi, A., Blonda, P., Parmiggiani, F., Pasquariello, G., Satalino, G.: Model Transitions in Descending FLVQ. IEEE Transactions on Neural Networks 9(5), 724–738 (1998)
12. Li, R.P., Mukaidono, M.: Gaussian Clustering Method Based on Maximum-Fuzzy-Entropy Interpretation. Fuzzy Sets and Systems 102(2), 253–258 (1999)
13. Ross, T.J.: Fuzzy Logic with Engineering Applications. McGraw-Hill, New York (1995)
14. Hand, D.J., Daly, F., Lunn, A.D., McConway, K.J., Ostrowski, E.: A Handbook of Small Data Sets. Chapman and Hall, London (1994)

A Fuzzy Approach to Skin Color Detection

Francisco A. Pujol, Rafael Espí, Higinio Mora, and José Luis Sánchez

Specialized Processor Architectures Lab
Dept. Tecnología Informática y Computación
Universidad de Alicante
P.O. Box 99, E-03080
Alicante (Spain)
fpujol@dtic.ua.es, reb2@alu.ua.es, hmora@dtic.ua.es,
sanchez@dtic.ua.es

Abstract. Skin detection is a useful tool for many subsequent image processing tasks, such as face detection, face tracking or human-computer interfaces. In many cases, using a skin color detection scheme is related to a proper representation of the color space chosen to interpret image information. In this work, we propose a fuzzy system for detecting skin in the RGB color space, so that each color plane is modeled using fuzzy sets. As a consequence, the development of the membership functions, the inference system and the defuzzifying process are described throughout this paper. The experiments have been performed with the XM2VTS and VALID face databases and the results show that our system has high detection rates and low false positive rates compared to other similar skin color detectors.

Keywords: Image processing, Skin Detection, Fuzzy systems, Machine learning.

1 Introduction

One of the most active research areas in computer vision is face recognition. Before a recognition algorithm is applied, faces must have been segmented and located using some detection method. Therefore, detecting a face is the first step for the recognition process to be efficiently completed. From this detection, one can utilize different methods to recognize users using facial features.

There are many approaches to face detection, such as knowledge- and appearance-based methods, feature invariant algorithms or template matching techniques (for further details, see [1], [2], [3]). In this work, we shall focus on a pre-processing stage for detecting faces in color images. To do this, we assume that the problem consists of detecting skin regions in the image, and then the segmented skin clusters will be further processed to validate they belong to face regions.

Many recent proposals are based on the underlying idea of representing the skin color in an optimal color space (such as RGB, YIQ or HSV) by means of the so-called skin cluster (see [2], [3], [4]). Color information is an efficient tool for identifying facial areas if the skin color model can be properly adapted for different lighting

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 532–542, 2008.

environments. This fact leads to avoid the use of RGB in practical systems, since the red, green and blue components are highly correlated and dependent on lighting conditions. However, the RGB space corresponds most closely with the physical sensors for colored light such as the cones in the human eye or red, green and blue filters in most color CCD sensors. In addition, using a RGB model should simplify the design of any algorithm, since there is no need to transform the color spaces.

As a consequence, our proposal is based on the RGB color space, using fuzzy sets instead of traditional color spaces; thus, input color variables are fuzzified and, as a result, a pixel is classified after the defuzzifying process is completed. To do this, in Section 2 we will revise some skin color detection algorithms and then the designing of the fuzzy system for detecting skin is described in Section 3. After that, the experiments are outlined and discussed in Section 4 and, finally, we shall conclude with some remarks to our work in Section 5.

2 Skin Color Detection

As described before, the main goal of skin color modeling is to define a decision rule that accurately separates the pixels of an image that correspond with skin from any other elements in a scene. There are a wide variety of methods to perform this task (see [1], [2] for a complete review); in this section some of them are shown.

In general, a method to develop a skin classifier consists of defining the bounding limits of the regions corresponding to a color which belongs to the skin, by means of some numerical (and often empirical) rules, i.e., defining explicitly the skin regions. It is clear that the efficiency of the color segmentation stage highly depends on the color space chosen. Color spaces with separated luminance and chrominance components, like HSV or YCbCr, seem to be appropriate for face detection (see, for instance, [1], [3], [4]). However, Albiol et al. [2] shown that if an optimum skin detector is designed for every color space, their performance will be almost the same.

The main advantage of this method is the simplicity of the skin detection rules, which permit a fast classification of pixels. The major drawback is the difficulty in finding both robust color spaces and reliable decision rules for a convenient skin detection. Nevertheless, for any color space, the problem to be solved can be summarized as: finding the optimal bounding limits for robust skin color detection.

On the other hand, there is a set of techniques which estimate the distribution of skin color by means of a training phase, without deriving an explicit model. These methods are often referred as non-parametric skin models. Generally, they create a skin probability map (SPM) [5], which consists of assigning a probability to each point in different color spaces. Finally, other methods include parametric skin distribution models, such as the Gaussian skin color model or the elliptic boundary model [1], [2].

Non parametric methods achieve low computation times for both training and classifying skin pixels, and they do not depend on the skin distribution shape or on the selection of the color space, either. However, they require a huge storage capacity and that the training must be performed with a representative dataset.

Parametric methods can also be fast, as they make it possible to interpolate data and to generalize the training phase for incomplete data. In spite of this, their

performance highly depends on the skin distribution shape and many parametric models ignore the static non-skin color. Subsequently, this fact leads to high false positive rates compared with non-parametric methods.

As a consequence, our proposal consists of taking the main advantages of explicitly defined skin models, i.e., the simplicity of their decision rules, and combining them with the main benefits of non-parametric skin models, that is to say, low computation times for training and classifying. To do this, our method includes a fuzzy model for detecting skin in color images. This model is explained in the following section.

3 A Fuzzy System for Detecting Skin

In any image segmentation scheme, there is a high level of uncertainty for a classifier to automatically obtain an optimum segmentation. This fact can be also extended to face detection and, in particular, to skin color segmentation. Thus, we find that applying fuzzy theory can be a convenient way to obtain good detection rates, since a fuzzy set-theoretic model provides a mechanism to represent and manipulate uncertainty within an image.

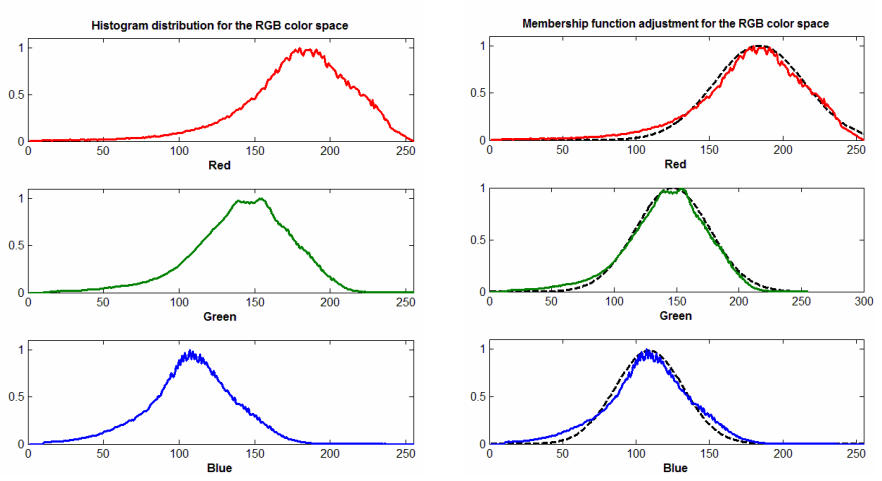
Color image segmentation using fuzzy classification is a pixel-based segmentation method. This method assigns a color class to each pixel of an input image by applying a set of fuzzy rules on it. We can use this approach to achieve our goal: a pixel can be classified as 'skin' or 'non-skin' according to a set of fuzzy rules extracted from a training stage using different color spaces. To do this, each color plane will be considered as a fuzzy set, so that the skin detection is performed through fuzzy functions representing the membership degree of each pixel to the different classes. Let us define some terms first.

Definition 1. A color vector $c \in C$, where C is a color space, of a pixel p is defined as a l-tuple of color components $c(p) = \{c_1(p), c_2(p),\ldots,c_l(p)\}$, where $c_i(p)$, for $i = 1,2,\ldots l$, may have N different values.

For instance, for the RGB color space, $l = 3$ and $c(p) = \{r_p, g_p, b_p\}$, where $r_p, g_p, b_p \in [0, 255]$.

Definition 2. Given a color image I of size $W = n \times m$ pixels, where each pixel is defined by a color vector c in a color space C, so that $c(p) = \{c_1(p), c_2(p),\ldots,c_l(p)\}$, $\forall\, p \in I$, the histogram of C, $H(C)$, is defined as a $q \times l$ array $H(C) = \{f_1, f_2, \ldots, f_l\}$, such that each f_i is the frequency vector, using q bins, of the color component c_i, for $i = 1,2,\ldots l$, on the image I.

As a result, the value of each bin is the number of pixels in image I having the color c_i. If $H(C)$ is normalized by W, then $H(C)$ takes the color space C into the interval [0, 1]; that is, $H(C)$ represents the probability distribution of each colour c_i to be present in image I. According to Zadeh's theory [6], a fuzzy set is a pair (A, m) where A is a set and m: $A \rightarrow [0, 1]$. This can be applied to the color histogram, where the fuzzy set can be defined as the pair (C, H), where C is the color space and H: $C \rightarrow [0, 1]$ is the



(a) (b)

Fig. 1. Creation of a fuzzy system for the RGB model. (a) Computation of $H(C)$ for the training set. (b) Gaussian approximation of $H(C)$ for each color plane.

normalized histogram. For each $c \in C$, $H(c)$ is the grade of membership of c, so that $c \in (C, H) \leftrightarrow c \in C \; AND \; H(c) \neq 0$.

Some previous works have modeled color spaces by means of fuzzy sets and relations. For the HSV space and its variants, the hue component has been defined through a fuzzy representation to take into account the non-uniformity of colors distribution. Thus, Truck et al. proposed to represent colors with trapezoidal or triangular fuzzy subsets, associating colors with fuzzy sets [7]; Herrera and Martinez use fuzzy linguistic hierarchies with different number of labels, depending on the desired granularity [8]. Some other approaches, such as [9], [10], use other common spaces for color-based classification, such as YCbCr or CIELab. However, there are little works dealing with fuzzy RGB classification and a technique for classifying skin color pixels in the RGB color space using a fuzzy approach has not been reported yet.

In order to use this fuzzy approach, we must calculate first the normalized histogram for the RGB color space for a training set, $H(C) = \{f_R, f_G, f_B\}$. Since the skin detection will be used as a pre-processing task for detecting a face in an image, for the training of the system let us consider two different face databases:

- The XM2VTS face database [11] contains 8 recordings of 295 subjects each, acquired over a period of 4 months. The background in this set of images is homogeneous, with a good contrast for detecting skin.
- The VALID database [12] consists of five recording sessions of 106 subjects over a period of one month. One session is recorded in a studio with controlled lighting and no background noise, the other 4 sessions are recorded in office type scenarios.

The training has been performed using a set of 100 images from both databases, extracting only the skin information, and using different ethnic groups and changing lighting conditions.

The results after obtaining $H(C)$ are shown in Fig. 1(a). From these results, the membership functions for the skin color in each plane can be modeled using a Gaussian function, such that:

$$\mu_{SKIN_i}(c_i) = \beta_i \, \mathrm{e}^{-\frac{(c_i-\alpha_i)^2}{2\sigma_i^2}} \quad . \tag{1}$$

where $i = \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$; $\{c_R, c_G, c_B\} \in [0, 255]$; $\beta_i = \max(H(c_i))$, $\alpha_i = \arg\max_{c_i} H(c_i)$, and σ_i^2 is the variance of each fuzzy set c_i. The results of the model for the skin pixels are shown on Fig. 1(b).

Finally, for the background pixels, i.e., the non-skin pixels in the image, let us consider a variation of the model introduced by Murthy and Pal [13], which identifies the fuzziness in the transition region between the object (in this case, the skin) and the background classes. Thus, the membership value of a point to the object is determined by applying an S-function and a Z-function to each color plane, so that:

$$\mu^S_{NON-SKIN_i}(c_i) = \begin{cases} 0 & \text{for } c_i \le a_{Si} \\ 2\left(\dfrac{c_i - a_{Si}}{\gamma_{Si} - a_{Si}}\right)^2 & \text{for } a_{Si} \le c_i \le b_{Si} \\ 1 - 2\left(\dfrac{c_i - \gamma_{Si}}{\gamma_{Si} - a_{Si}}\right)^2 & \text{for } b_{Si} \le c_i \le \gamma_{Si} \\ 1 & \text{for } \gamma_{Si} \le c_i \end{cases} . \tag{2}$$

$$\mu^Z_{NON-SKIN_i}(c_i) = \begin{cases} 1 & \text{for } x \le a_{Zi} \\ 1 - 2\left(\dfrac{x - a_{Zi}}{\gamma_{Zi} - a_{Zi}}\right)^2 & \text{for } a_{Zi} \le x \le b_{Zi} \\ 2\left(\dfrac{x - \gamma_{Zi}}{\gamma_{Zi} - a_{Zi}}\right)^2 & \text{for } b_{Zi} \le x \le \gamma_{Zi} \\ 0 & \text{for } \gamma_{Zi} \le x \end{cases} . \tag{3}$$

where the values b_{Si}, b_{Zi} are the cross-over points of the fuzzy sets defined by $\mu^S_{NON-SKIN_i}(c_i)$, $\mu^Z_{NON-SKIN_i}(c_i)$, respectively, for $i = \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$; i.e., the membership value of $c_i = b_{Si}$ (or b_{Zi}) that is equal to 0.5; $\{c_R, c_G, c_B\} \in [0, 255]$; $a_{Si} = \gamma_{Zi} = \arg\max_{c_i} H(c_i)$; $a_{Zi} = 0.5 \cdot \gamma_{Zi}$; and $\gamma_{Si} = 1.5 \cdot a_{Si}$. The results for the models of the fuzzy skin and non-skin classes are shown in Fig. 2; we have three classes for each color fuzzy set.

Now, given an input image I, for any pixel $p \in I$, its R, G, B components are fuzzified, according to the parameters defined in Eqs. (1)-(3). Then, the inferencing system processes each pixel and, by using the available knowledge in the form of IF-THEN rules, it identifies and classifies skin color pixels in the output image. This procedure

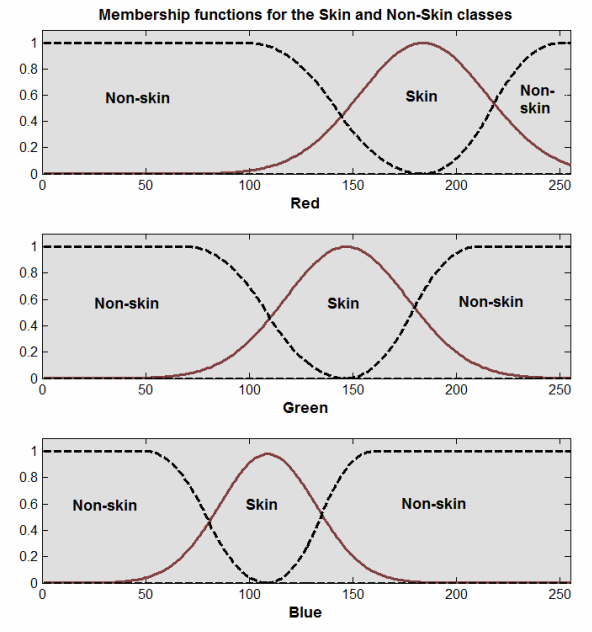


Fig. 2. Modeling the Skin and Non-skin Classes for the RGB color space

results in the assignment of one output fuzzy set for each rule. A total of 12 rules were extracted for our system. The min–max inferencing technique was used, where the output membership function of each rule is clipped off at a height corresponding to the rule premise's computed degree of truth. The combined fuzzy output membership function is built by combining the results of all the fuzzy rules. If an output fuzzy set is activated by more than one rule, the maximum of all activations is considered in the construction of the combined output membership function.

The Mamdani method was chosen as the defuzzification procedure, which means that the fuzzy sets obtained by applying each inference rule to the input data were joined through the add function; the output of the system was then computed as the centroid of the resulting membership function.

After the fuzzy inference system has been defined, in the following section we describe the results of the experiments performed for the different databases.

4 Experimental Results

Let us show now the results of some experiments completed for our model and compare it with some other existing methods. The tests have been performed with the XM2VTS (homogeneous background) and the VALID (homogeneous and more complex backgrounds) face databases, as described in Section 3.

According to the fuzzy skin detector defined before, Fig. 3 shows the results for different images in both databases. The images include different gender and ethnic

Fig. 3. Results of the fuzzy skin detector for different images. First row: some images from the XM2VTS database. Second row: some images from the VALID database.

groups, and –for the VALID database– complex backgrounds and changing lighting conditions.

From these results, we must point out that the proposed method leads to very accurate results for the XM2VTS database, which has both good illumination conditions and high contrast between the foreground and the background. In the case of the VALID database, the results of the skin detector are also quite precise, in spite of the unfavorable conditions for achieving good quality results. Some errors appear, mainly due to background colors that are very similar (or even identical) to skin colors; in addition, the lighting conditions will affect the final result. However, with a post-processing stage for detecting a face from the skin segmentation results, most of these errors will be corrected.

Finally, let us compare our system with some other existing algorithms for skin color detection. First, an explicitly defined skin region algorithm for the RGB color space is used; in [14], a pixel (R_0, G_0, B_0) is classified as *skin* if:

$$\begin{aligned} &\text{(i)}\ \ R_0 > 95 \wedge G_0 > 40 \wedge B_0 > 20, \ \text{and} \\ &\text{(ii)}\ \ \max(R_0, G_0, B_0) - \min(R_0, G_0, B_0) > 15, \ \text{and} \\ &\text{(iii)}\ \ |R_0 - G_0| > 15 \wedge R_0 > G_0 \wedge R_0 > B_0. \end{aligned} \tag{4}$$

On the other hand, a skin probability map (SPM) approach to the RGB space is also taken into account. For this method, given skin and non-skin histogram models, a skin pixel classifier is defined using, for instance, the Bayes' theorem. Then, the inequality $P(skin|c) \geq T$ can be used as a skin detection rule, where T is a selected threshold value and $P(skin|c)$ is the probability of observing skin, given a concrete c color value (see [1], [5] for further details). For completion, we have included SPMs for the HSV and YCbCr spaces, as well.

Figs. 4-6 show the results for some images for the proposed and existing segmentation methods. In every case, for the SPM models, we show the threshold T that minimizes the number of false positives and false negatives.

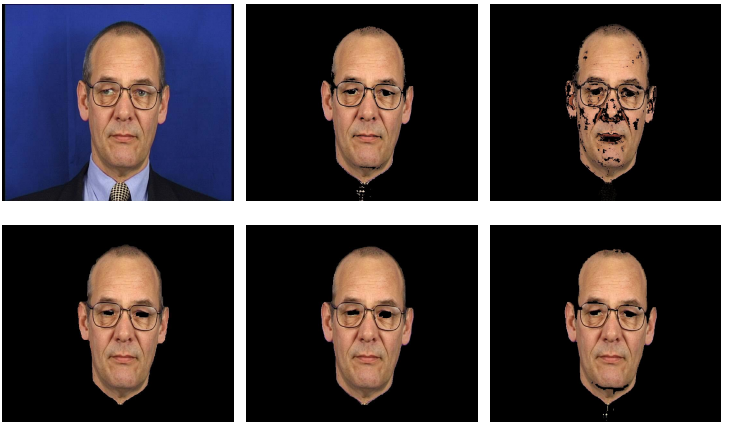


Fig. 4. Results from the XM2VTS database. From left to right, top to bottom: original image; results for explicit RGB rules; results for SPM in RGB, $T = 0.12$; results for SPM in HSV, $T = 0.06$; results for SPM in YCbCr, $T = 0.04$; results for our model.

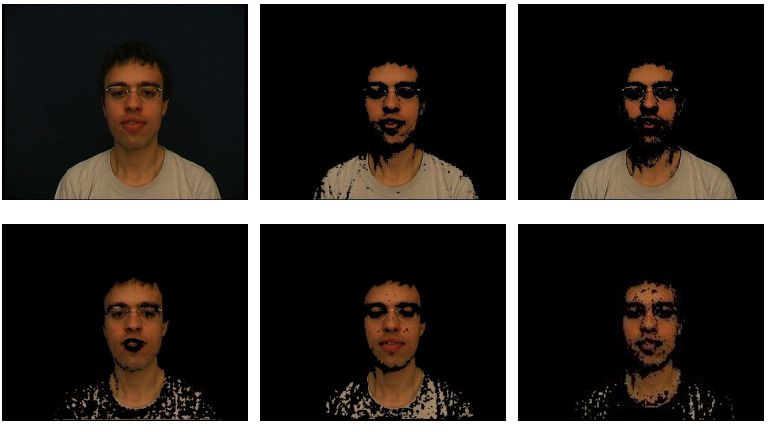


Fig. 5. Results from the VALID database. From left to right, top to bottom: original image; results for explicit RGB rules; results for SPM in RGB, $T = 0.25$; results for SPM in HSV, $T = 0.37$; results for SPM in YCbCr, $T = 0.43$; results for our model.

Finally, in Table 1 we show the results of the false positive (FP) and false negative (FN) rates, and correct skin detection rate for the images in the experiments, for all the models considered in our implementation.

The results confirm the validity of the fuzzy RGB skin detector; the global results achieve a detection rate of 94.60% (the highest one from the compared methods), with only a 1.73% of false positives (the lowest one from the compared methods). The performance is better than the other compared RGB proposals; when there is a homogeneous background, with good lighting conditions, almost any skin detector gives precise results, but when we use unconstrained environments, it is more difficult

Fig. 6. Results from the VALID database with a complex background. From left to right, top to bottom: original image; results for explicit RGB rules; results for SPM in RGB, $T = 0.15$; results for SPM in HSV, $T = 0.05$; results for SPM in YCbCr, $T = 0.03$; results for our model.

Table 1. Comparison between algorithms

	XM2VTS			VALID			Global Results		
Algorithm	FP(%)	FN(%)	Correct(%)	FP(%)	FN(%)	Correct(%)	FP(%)	FN(%)	Correct(%)
Explicit RGB rules	2.13	3.07	94.80	4.67	9.28	86.05	2.97	5.14	91.89
SPM RGB	3.67	3.20	93.13	8.74	13.55	77.71	5.36	6.65	87.99
SPM HSV	1.62	3.09	95.29	4.38	8.07	87.55	2.54	4.75	92.71
SPM YCbCr	2.51	2.61	94.88	3.46	5.74	90.80	2.82	3.65	93.53
Fuzzy RGB	1.54	2.73	95.73	2.11	5.57	92.32	1.73	3.67	94.60

to get a good skin segmentation. This fact can be clearly noticed for the VALID database and the RGB space for both the explicitly defined regions and the SPM methods, where many false positives and negatives appear and the correct detection rates are far from being optimal, especially for the SPM approach. Unlike the other RGB proposals, our method behaves well for complex backgrounds and changing illumination, giving very accurate results.

On the other hand, the FP and FN rates show that the fuzzy RGB skin detector has a comparable behavior than other methods in color spaces traditionally used for skin color detection, such as HSV or YCbCr. From Fig. 7 it can be extracted that when there is a strong presence of a red object, almost any skin segmentation method gives false positives for the pixels in the object; although the HSV and YCbCr spaces provide a good segmentation result, they can't remove many of the red pixels from the objects in the background, whereas the fuzzy RGB achieves a more appropriate result. However, there are some situations with unconstrained backgrounds where it is rather difficult for any skin detector to generate a proper segmentation. Thus, the performance of the last three algorithms in Table 1 is comparable, so that any of them can be

used as a pre-processing stage for more complex algorithms, such as a face recognition system or a human-machine interface.

To sum up, the fuzzy system has high correct detection rates and, as a conclusion, our method is a suitable technique for segmenting skin in different environment conditions, such as complex backgrounds or changing lighting conditions.

5 Conclusions

Face detection is object of intensive research in the last few years and this tendency is increasing, since it is the initial and, perhaps, the critical process for an integral face recognition system. One of the methods used for detect a face is to use a skin segmentation scheme, which has been proved to be highly effective in many applications.

In this work, a fuzzy skin color detector using the RGB color space has been proposed. Each color plane has been modeled using different fuzzy sets for the skin and non-skin classes and the inference system and defuzzifying process have been exposed. Experiments show that our method leads to very accurate skin detection results with low false positive and false negative rates.

As a future work, we propose to extend the fuzzy system to other color spaces, such as HSV or YCbCr. Then, some algorithm for detecting a face in an image must be developed, as well. It would be also desirable to use our algorithm with other databases and in a real environment to test the validity of our fuzzy system proposal.

References

1. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002)
2. Vezhnevets, V., Sazonov, V., Andreeva, A.: A Survey on Pixel-Based Skin Color Detection Techniques. In: Proc. Graphicon 2003, Moscow, Russia, September 2003, pp. 85–92 (2003)
3. Albiol, A., Torres, L., Delp, E.J.: Optimum color spaces for skin detection. In: Proc. ICIP 2001, Thessaloniki, Greece, pp. 122–124 (2001)
4. Yang, J., Fu, Z., Tan, T., Hu, W.: Skin Color Detection Using Multiple Cues. In: Proc. ICPR 2004, Cambridge, UK, August 23-26, 2004, vol. I, pp. 632–635 (2004)
5. Brand, J., Mason, J.S., Roach, M., Pawlewski, M.: Enhancing Face Detection in Colour Images using a Skin Probability Map. In: Proc. of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, China, pp. 344–347 (2001)
6. Zadeh, L.A.: Fuzzy Sets. Information and Control 8(3), 338–353 (1965)
7. Truck, I., Akdag, H., Borgi, A.: A Symbolic Approach for Colorimetric Alterations. In: Proc. of EUSFLAT 2001, Leicester, England, pp. 105–108 (2001)
8. Herrera, F., Martínez, L.: A Model based on Lingustic two-tuples for Dealing with Multigranularity Hierarchical Linguistic Contexts in Multiexpert Decision-Making. IEEE Transactions on Systems, Man and Cybernetics 31(2), 227–234 (2001)
9. Vertan, C., Boujemaa, N.: Using Fuzzy Histograms and Distances for Color Image Retrieval. In: Proc. of CIR 2000, Brighton, United Kingdom (2000)
10. Guironnet, M., Pellerin, D., Rombaut, M.: Video Classification Based on Low-Level Feature Fusion Model. In: Proc. of EUSIPCO 2005, Antalya, Turkey (2005)

11. Messer, K., Matas, J., Kittler, J., Luettin, J., Maitre, G.: XM2VTSDB: The Extended M2VTS Database. In: Proc. Second International Conference on Audio and Video-based Biometric Person Authentication, Washington D.C, March 1999, pp. 72–77 (1999)
12. Fox, N.A., O'Mullane, B.A., Reilly, R.B.: VALID: A New Practical Audio-Visual Database, and Comparative Results. In: Kanade, T., Jain, A.K., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 777–786. Springer, Heidelberg (2005)
13. Murthy, C.A., Pal, S.K.: Fuzzy Thresholding: Mathematical Framework, Bound Functions and Weighted Moving Average Technique. Pattern Recognition Letters 11, 197–206 (1990)
14. Kovač, J., Peer, P., Solina, F.: Human Skin Colour Clustering for Face Detection. In: Proc. EUROCON 2003, Ljubljana, Slovenia, September 22-24, 2003, vol. 2, pp. 144–148 (2003)

To Boldly Split: Partitioning Space Filling Curves by Markov Chain Monte Carlo Simulation

Daniel Valdes and Abhir Bhalerao

Department of Computer Science
The University of Warwick
Coventry, UK, CV4 7AL
{dvaldes,abhir}@dcs.warwick.ac.uk

Abstract. Space filling curves are a class of fractals that are important mathematical descriptions of the appearance and shape of natural objects. There is growing interest in the modelling of such curves to measure pathology in medicine and biology. This work presents a method of modelling fractal curves, such as the boundary of brain white matter, and partitioning such curves in to segments having equal fractal dimension. Since the solution space, for a given number of contour points and a required set of partitions is very large, we employ a Bayesian framework of reversible-jump Markov chain Monte Carlo (MCMC) and a sampler based on the Metropolis-Hastings test. We detail the algorithm and present results on both simple contours (animal silhouettes) and space-filling brain contours and show the convergence characteristics of the method. We discuss its use for building compact local statistical shape models.

1 Introduction

Being able to partition a contour into a set of meaningful local parts has many applications in machine vision, image retrieval, terrain classification, handwriting recognition etc. Indeed, certain types of curves, e.g. fractals, can only be characterised by their local properties and are either repetitive arrangements of similar patterns or the result of applying a syntactical rule over a set of scales.

Mandelbrot's [1] work on fractal geometry demonstrated how fractal dimension could characterise many natural phenomena, such as the folding of the human brain cortex, and it has motivated clinical studies that use 3D MRI to understand the folding process and correlate changes in fractal dimension with age and disease [2,3,4,5]. In [2], a study reported the effects of age related white matter structural changes on estimates of fractal dimension. A similar study in [3] performed a three-dimensional fractal analysis on the white matter surface to quantify its complexity. A volumetric method based on the FFT was used in [4] to address the question of whether the human cerebral cortex is indeed self-similar. Kontos et al. [5] provided a study of MRI activation patterns using space filling curves to relate brain structures with brain functions.

Brain imaging has empowered medicine and neuroscience with the ability to use the images to identify structural brain changes associated to neurodegenrative diseases. Since the brain structure is so complex and varies across subjects, methods to identify

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 543–553, 2008.

morphological variability can be useful for diagnosis, treatment and to assess the structural basis of normality and disease [6]. Statistical shape modelling (SSM) is concerned with the analysis of shapes of sub-structures and aims to describe variations across groups of individuals (taking into account for example age, sex, genetic background and disease state). Shape modelling has also proven that brain variation can be successfully captured [7], due to the effectiveness of capturing the variation of sets of contours (shapes) [8]. By this approach, a set of training shapes is used to build the model and then a given test-shape can be compared against the model. Fractals and space filling curves defy such analysis because each example is, so to speak, its own exemplar and the relative complexity of such shapes: (1) make it hard to establish meaningful correspondences between shapes; and (2) require a much larger training size. This reflects that work on shape modelling is constrained by many unsolved problems including segmentation of structures, dependency on manual data processing and difficulties in modelling *local* versus *global* variation. A local SSM approach was proposed in [9] and it was demonstrated how local deviations from the fractal characteristics of space-filling curves could be analysed based on pose-alignment and an SSM. This work allows the pose variation between symmetrical parts of the curve to be eliminated and, in theory, allows the SSM to represent the interesting variation between parts. However, an open question of this work was what should be a good partition of a given contour into local parts and how can this be determined in a simple and efficient way.

Several methods for contour partitioning have been reported in the image analysis literature. Richards and Hoffman [10] developed shape descriptors known as codons that decompose contours at negative curvature extremes. Asada and Brady [11] introduced a method based on Marr's primal sketch representation of intensity changes in gray level images to describe significant changes in curvature by matching the multi-scale convolutions of a shape. Baruch et al. [12] reported a method that uses the differential chain code of the edge to segment it into sections with constant curvature. Furthermore, Wuescher and Boyer created a technique for partitioning contours into constant curvature segments using a nonlinear blip filters matched to the impairment signature of the curvature computation [13]. More recently Cronin [14] proposed a method based on parsing concavity to partition digital contours into concave and convex sections. The use of Curvature Scale-Space (CSS) as proposed by Mokhtarian [15] and later applied by Abbasi et al [16] in their work on contour modelling for retrieval (the University of Surreys SQUIDS retrieval-by example database), is also an important and useful development.

In this paper, we address the problem of partitioning contours to support the construction of local statistical shape models, where the contour partitioning needs to be guided by the fractal dimension. We show that the resulting algorithm is also able to find symmetries in curves, which may be of use in other applications. Given a closed contour of N points and M being the number of required partitions, if $M \ll N$, it is not hard to see that the number of potential partitions becomes very large[1]. Because of the need to search a large solution space, the problem is cast into a Bayesian

[1] For example: if $N = 512$ and $M = 16$, then in the case that each partition starts with 32 points, and we allowed neighbouring partitions to exchange $\frac{1}{2}$ their points by moving the border, there would be $16^{16} \approx 2^{19}$ possible configurations to test.

framework by setting up a stochastic search within the parameter space configurations of M required partitions. The proposed method draws samples from a parameter space using a reversible-jump Markov chain Monte Carlo simulation [17] where each state of the chain represents a likely partition and can be assessed using a posterior probability. In the following, we describe in detail how the contour partitioning problem can be cast into this framework and illustrate its operation on a set of simple object outlines (animal silhouettes) and brain white matter contours. We conclude by discussing the potential of this approach.

2 Markov Chain Monte Carlo Simulation for Contour Partitioning

Let a piece-wise linear contour of N points be defined by the coordinates: $X = \{\overrightarrow{x}_i\}$, $i = 1 \ldots N$, and let a given partition of M exclusive subsets be the parameter sets $\Theta = \{\Theta_1, \Theta_2, \ldots, \Theta_M\}$. Let the posterior probability of a given partition be given by

$$P(\Theta|X). \tag{1}$$

Then by Bayes' rule, $p(\Theta|X) \propto p(X|\Theta)p(\Theta)$, which is the product of the likelihood of the data given a partition Θ and the *prior probability* of the partition Θ.

As the aim of the partitioning process is to end up with λ_M contour subsections Θ_k of more or less equal size $(card[\Theta_j])$ and equal fractal dimension, we can

1. Model the prior as
$$\pi_n(k; \Theta_j) = \frac{e^{-\lambda_n}\lambda_n^k}{k!}, \tag{2}$$
which is a Poisson density with rate parameter λ_n.
2. Model the data distribution by the likelihood function
$$l(\Theta_j|X) = e^{-|FD(\Theta_j)-t|}, \tag{3}$$
which favours the partition having a fractal dimension, FD, of t.
3. Penalise the posterior by a density to keep the number of partitions to be the desired number, $\lambda_M = M$, the required partition size,
$$\pi_M(k; \Theta) = \frac{e^{-\lambda_M}\lambda_M^k}{k!}. \tag{4}$$

Taken together, we can now expand the posterior (1) as

$$p(\Theta|X) \propto \pi_M(card[\Theta]; \Theta) \prod_j^M l(\Theta_j|X)\pi_n(card[\Theta_j]; \Theta_j) \tag{5}$$

In the simulations presented below, for the likelihood function we use a value of $t = 2.57$ based on studies of brain contours. Note that this probability model is being setup to assume that there are *local* parts of the contour which have fractal dimension t. Of course, if it were a fractal and not discretely sampled then by definition this assumption would hold at arbitrary scales.

2.1 Stochastic Sampling and Reversible Jumps

Markov chain simulation (or MCMC) is a method based on drawing values of parameters Θ from functions that approximates it, and then draws are altered to obtain a better approximation to the target posterior distribution $p(\Theta|X)$. Since the samples are drawn sequentially and the distribution that generates these depends only on the previous value drawn, the set of states form a Markov chain [18]. To ensure the success of the algorithm it is necessary to create a Markov process whose stationary distribution is the specified $p(\Theta|X)$ and then run the simulation for a sufficient length of time.

The Metropolis-Hastings (MH) algorithm can be considered as a generalisation of a class of Markov chain simulation methods for drawing samples from posterior distributions. Given a candidate-generating density, $p(\Theta, \Theta')$, of moving from state Θ to Θ' , it is clear that the stationary distribution is reached when $p(\Theta'|X)p(\Theta, \Theta') = p(\Theta|X)p(\Theta', \Theta)$ is satisfied for all Θ, Θ'. The MH algorithm performs rejection sampling by comparing the validity of the two states by forming an acceptance ratio

$$\alpha(\Theta, \Theta') = min\left[1, \frac{p(\Theta'|X)p(\Theta, \Theta')}{p(\Theta|X)p(\Theta', \Theta)}\right],$$

and then, if $\alpha = 1$, the move $\Theta \longrightarrow \Theta'$ is accepted, otherwise it is accepted with probability $\alpha \geq r \sim U[0, 1]$. Note that it is easy to see that if the candidate-generating densities are equal the state changes $\Theta \longrightarrow \Theta'$ and $\Theta' \longrightarrow \Theta$, then the chain automatically moves to a higher probability state. For the contour partitioning, we allow three moves in the state space: splitting a chosen contour partition at a random point; merging a random pair of neighbouring partitions and altering the position of a boundary between neighbouring segments (see figure 1). The simulation is initialised with a random number of partitions (e.g. 12) and equal candidate-generating probabilities: $P(split) = P(merge) = P(alter) = \frac{1}{2}$, with moves chosen at random. Note that for the split move, the random position is first chosen in the range $1 \ldots N$, then the (current) segment in which it lies is split at that point. This ensures that larger segments are more likely to be split than smaller ones.

2.2 Estimating Fractal Dimension

The fractal dimension of a curve estimates of how much space it occupies relative to its length. For a planar curve, it can be calculated by the number, $N(\delta)$, of area

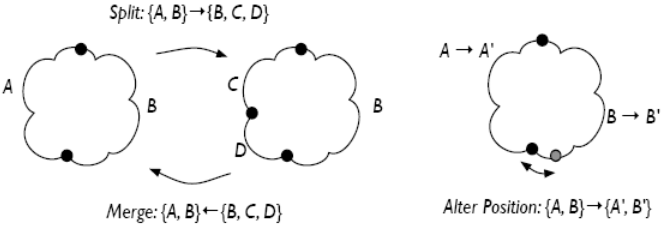


Fig. 1. Illustration of the (1) Split, Merge and (2) Alter Position moves

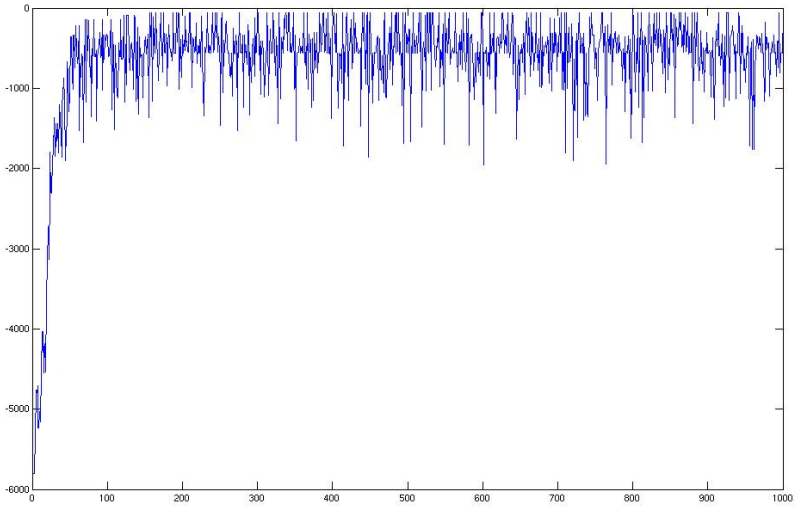


Fig. 2. Typical convergence characteristics. Plot shows log posterior against iteration (move) number. (This plot is for the brain white matter contour 'slice 179' used in the experiments below).

elements, δ, needed to cover the curve. The estimate, then, of the length of the curve is $N(\delta)\delta$. The Hausdorff-Besicovitch dimension D of a curve is defined as some measure $M_d(\delta)$ for which $N(\delta)\delta^d$ abruptly changes from zero to infinity [19]. Since D is often finite, it can be shown that for sufficiently small δ

$$D = -\frac{logN(\delta)}{log(\delta)}. \tag{6}$$

For practical purposes, by taking a set of meshes of varying δ over the curve, it is possible to count $N(\delta)$ and plot $logN(\delta)$ against $log(\delta)$.

2.3 Practical Considerations

To keep the posterior calculations numerically stable, they are performed in the log domain. Taking natural $logs$ of (5) we can replace the products by sums of log likelihoods and log Poisson densities. Also, for small M, the factorial of $(card[\Theta_j])$ can become large, so it is convenient to use Stirlings approximation of factorial,

$$ln\, n! \approx n[ln\, n - 1], \tag{7}$$

thus:

$$\begin{aligned} ln\, p(\Theta|X) \propto\ & m(ln[\lambda_M] - ln[m] - 1) - \lambda_M \\ & - \sum_{j}^{M} |fd(\Theta_j) - t| \\ & + \sum_{j} n_j(ln[\lambda_n] - ln[n_j] - 1) - \lambda_n, \end{aligned} \tag{8}$$

where $n_j = card[\Theta_j]$ and $m = card[\Theta]$.

As with all MCMC implementations, trial and error was used to determine at what number of steps convergence was being achieved. For the contours presented below (each with approximately 3000 points) we began by running 1000 steps, however convergence was observed after about 300 steps (see figure 2). Also, we manually selected one point on the contour as a fixed point which marked a line of symmetry: for all contours, this was one of the two points along the principal axis of the contour.

3 Experimental Results

We present results of contour partitioning using MCMC simulation on real contours: animal silhouettes (birds, fish, squids) and brain white matter outlines. The contours were sampled with approximately 3000 points and we chose to vary the number of required partitions, λ_M e.g. 2, 4, 8 etc. and set the mean number of points per partition, $\lambda_n = N/\lambda_M$. This required number of partitions (2, 4 and 8) where only used with illustrative purposes to demonstrate the ability of the algorithm to find symmetries, but others can be used. The simulations were initialised with M = 12 and $P(split) = P(merge) = P(alter) = \frac{1}{3}$ and a single point was user selected along a natural symmetry line. Each of the figures show the starting configuration of random partitions and the converged results. The plots of the resulting partitions present a colour

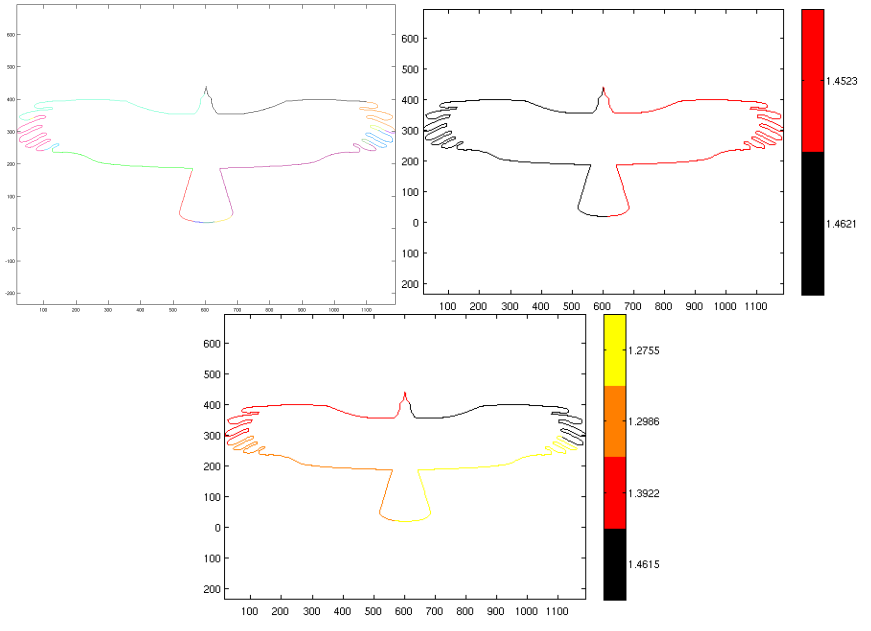


Fig. 3. Example results on bird contours: random starting partitions (top left), 2 (top right) and 4 (bottom) partitions respectively

Table 1. Results for the brain contour $\#\mathbf{179} : \boldsymbol{\lambda_M} = \mathbf{8}$

Partition	1	2	3	4	5	6	7	8
Points	378	455	441	432	440	403	380	366
FD	1.7107	1.5794	1.6357	1.6138	1.5362	1.6208	1.4745	1.5955

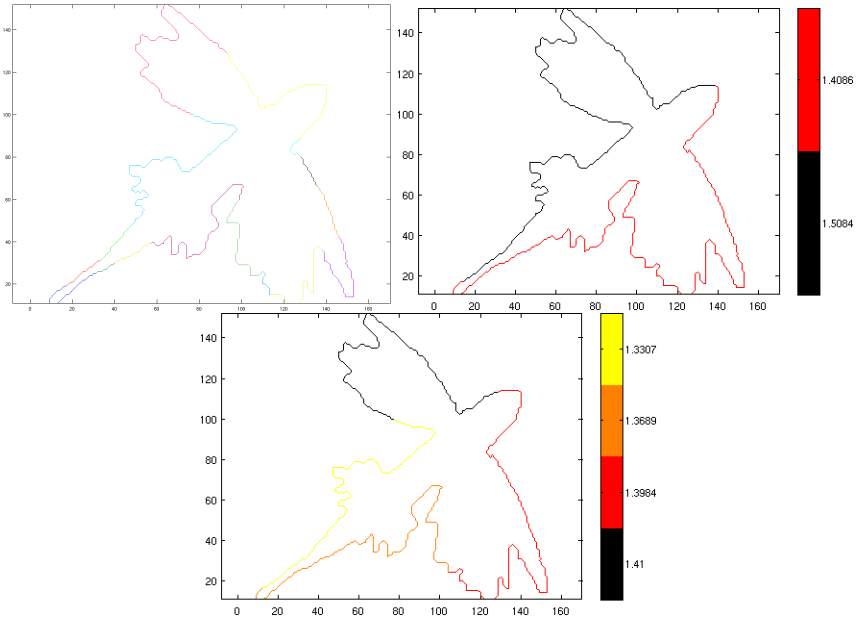


Fig. 4. Example results on SQUID database contours: random starting partitions (top left), 2 (top right) and 4 (bottom) partitions respectively

code going from black for the partition with the highest fractal dimension, to yellow, depicting the lowest.

For the bird (figure 3) and squid contour (figure 4), the data clearly has a one-fold symmetry along its length and we would expect this to be found. The results for 2, and 4 partitions are depicted. The partition into 2 is almost perfect in both cases. In the case of the squid, the symmetry is close to that required on the right-hand side of the head. These are relatively low curvature elements and the fractal dimension will not change much in moving the partition boundary from the centre of the head to the converged point. The results for 4 partitions are less obvious but it is encouraging to see that the partitioning is mirrored along the principal axis in both cases.

Figures 5 and 6 show results on a contour taken from white matter segmentation of two slices of a MRI brain image. Here, the plots of the resulting partitions present a colouring code going from black for the partition with the highest fractal dimension, to yellow, depicting the lowest. Tables 1 and 2 show the partitioning results in a quantitative way, describing the number of constituent points and the fractal dimension of the

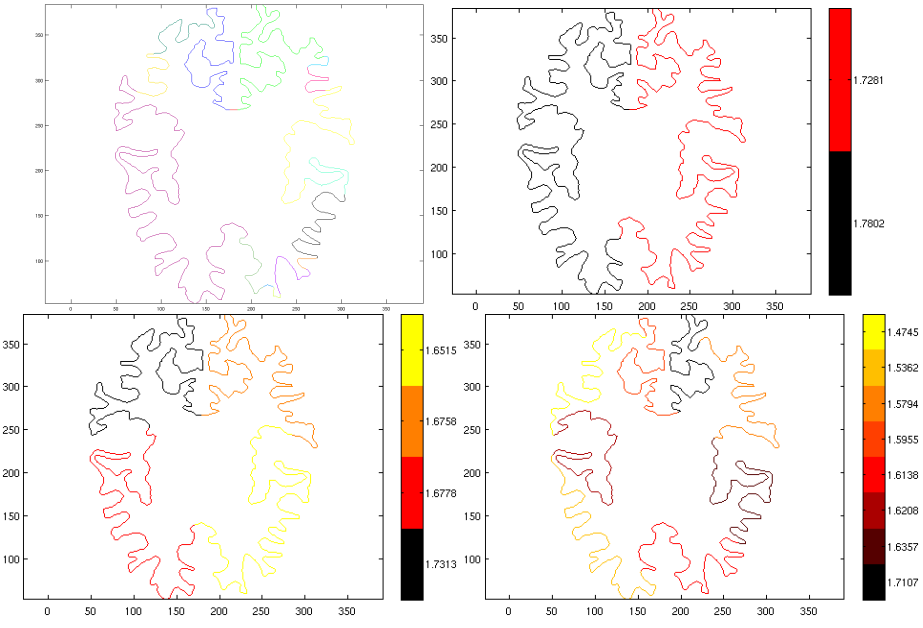


Fig. 5. White matter brain contour partitioning - slice $\#\mathbf{179}$. Top left: start configuration of MCMC using arbitrary partitioning. Results shown for $\boldsymbol{\lambda_M = 2}$ (top right), $\boldsymbol{\lambda_M = 4}$ (bottom left) and $\boldsymbol{\lambda_M = 8}$ (bottom right) partitions after 1000 iterations.

Table 2. Results for the brain contour $\#\mathbf{182} : \boldsymbol{\lambda_M = 8}$

Partition	1	2	3	4	5	6	7	8
Points	325	343	361	394	430	406	435	424
FD	1.6689	1.5502	1.5259	1.6208	1.5316	1.5929	1.6689	1.5973

resulting partitions. The contours are similar as they are taken from brain slices which are close by to each other (slices $\#179$ and $\#182$). The MCMC partitioning was run for 1000 steps and 2, 4, and 8 partitions were sought starting from 15 random segments. A point on the line which separate the left and right hemispheres of the brain, (on the sagittal line) was marked by hand. The symmetries of the two halves, and the anterior and posterior lobes are correctly found in both case ($\lambda_M = 2$ and $\lambda_M = 4$). The result when 8 partitions are sought is harder to interpret. For slice $\#179$ only 6 partitions are found at convergence, whereas for slice $\#182$, 8 were found. This illustrates that in the former, the likelihood is sufficiently strong so as to drive the result away from the Poisson rate of $\lambda_M = 8$, where as in the latter, 8 partitions is the better solution. Nevertheless, there is good agreement in both cases for the other partitions. If we were to use this to partition all the brain slices, one way to achieve better correspondence between slices would be to initialise neighbouring slices with converged partitions from the current slice.

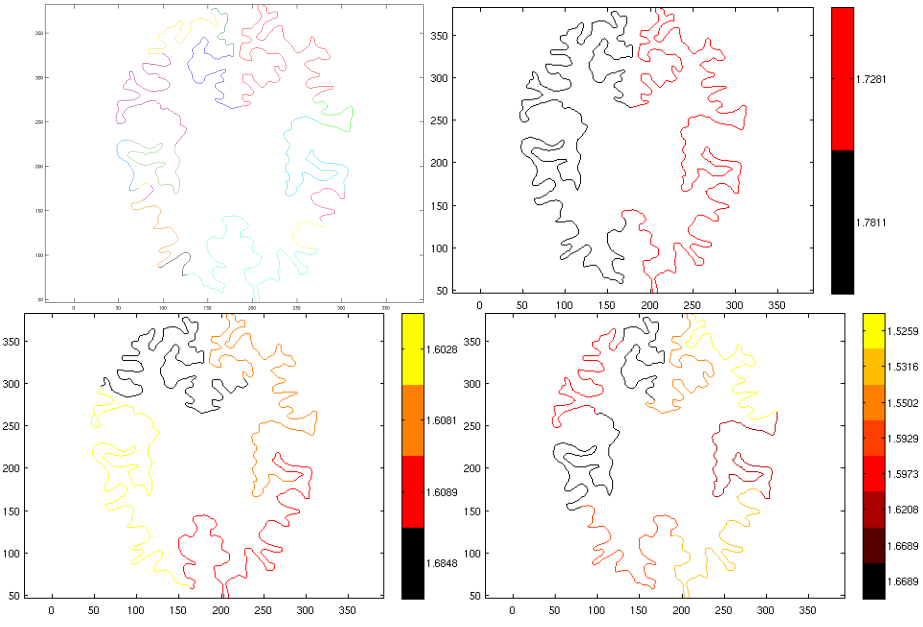


Fig. 6. White matter brain contour partitioning - slice **#182**. Top left: start configuration of MCMC using arbitrary partitioning. Results shown for $\boldsymbol{\lambda_M = 2}$ (top right), $\boldsymbol{\lambda_M = 4}$ (bottom left) and $\boldsymbol{\lambda_M = 8}$ (bottom right) partitions after 1000 iterations. Compare with figure 5.

4 Discussion

For a genuine fractal curve, by definition, any arbitrary partition should have the same fractal dimension. Even contours found in biology are not always fractals but can exhibit self-similarity, they must be discretely sampled and they may be made up of parts which are fractal and non-fractal. Brain contours are space filling (in 2D and 3D) and the presented method shows potential in being able to produce a reasonable set of partitions. Ultimately, the veracity of the result can only be confirmed by the construction of the shape models. In [20] it is shown how the resulting sets from contour partitioning are useful for analysis of local variation in brain structures.

The aim here is to be able to partition the contour into a set of objects which are related by affine symmetry and then variations from these local symmetries can be identified [9]. Moreover, another objective of the idea of generating sets of partitions from contours is to establish a way of determining meaningful local sets of shapes. We believe that that this model could have useful application in brain morphometrics and computational anatomy. To be more precise, the provided method can be adapted for clinical diagnosis software for assessing changes in local shape variation of anatomical structures, such as, white/gray matter. Finally, is possible to address the problem of image database retrieval where the objective can be to discover images which contain objects similar to query objects, in this case brain sections.

The proposed MCMC simulation approach has some important characteristics. It is only necessary to set the number of required partitions and the simulation is able effectively search the state space and adapt locally to shrink and grow to equalise the fractal dimension over the whole contour. This would not be possible to do exhaustively for nontrivial numbers of segments. By using a Bayesian framework and a stochastic sampler, the convergence is theoretically guaranteed: in practice, our experiments show convergence after a relatively modest number of moves. It is evident that other shape descriptors can be used to generate the partitions, like in [20] where using curvature scale space consistency is used to partition the the contour into locally similar parts. Here, the idea is exploring alternative solutions to this problem in the fashion of the Bayesian paradigm and to gain insight about the complexity of the brain structure.

There is scope to model the likelihood function in different ways. We chose to use fractal dimension but mean curvature could be a viable alternative. Also, our density uses an absolute norm $|\mathrm{fd}(\Theta_j)\text{-}t|$ and perhaps a Gaussian could be used instead. As well as investigating these questions, we are conducting analysis of several sets of brain contours and will be comparing our results with an implementations of a curvature scale space based method. An unsupervised selection of the initial point can be envisaged using the zero-crossings/extrema points generated by the contour evolutions of the CSS method [15].

References

1. Mandelbrot, B.B.: The fractal geometry of nature. Freeman, New York (1982)
2. Zhang, L., Liu, J.Z., Dean, D., Sahgal, V., Yue, G.H.: A three-dimensional fractal analysis method for quantifying white matter structure in human brain. Journal of Neuroscience Methods 150(2), 242–253 (2006)
3. Free, S.L., Sisodiya, S.M., Cook, M.J., Fish, D.R., Shorvon, S.D.: Three-dimensional fractal analysis of the white matter surface from magnetic resonance images of the human brain. Cerebral Cortex 6(6), 830–836 (1996)
4. Kiselev, V.G., Hahn, K.R., Auer, D.P.: Is the brain cortex a fractal? NeuroImage 20(3), 1765–1774 (2003)
5. Kontos, D., Megalooikonomou, V., Ghubade, N., Faloutsos, C.: Detecting Discriminative Functional MRI Activation Patterns Using Space Filling Curves. In: Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology, vol. 1, pp. 963–966 (2003)
6. Ashburner, J., Csernansky, J.G., Davatzikos, C., Fox, N.C., Frisoni, G.B., Thompson, P.M.: Computer-assisted imaging to assess brain structure in healthy and diseased brains. Lancet Neurology 2, 79–88 (2003)
7. Styner, M., Gerig, G., Lieberman, J., Jones, D., Weinberger, D.: Statistical shape analysis of neuroanatomical structures based on medial models. Medical Image Analysis 7, 207–220 (2003)
8. Cootes, T.F., Hill, A., Taylor, C.J., Haslam, J.: Use of active shape models for locating structures in medical images. Image and Vision Computing 12(6), 355–365 (1994)
9. Bhalerao, A., Wilson, R.: Warplets: An image-dependent wavelet representation. In: International Conference on Image Processing, ICIP, vol. 2, pp. 490–493 (2005)
10. Richards, W., Hoffman, D.D.: Codon constraints on closed 2-D shapes. Computer Vision, Graphics, and Image Processing 31(3), 265–281 (1985)

11. Asada, H., Brady, M.: Curvature primal sketch. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(1), 2–14 (1986)
12. Baruch, O., Loew, M.H.: Segmentation of two-dimensional boundaries using the chain code. Pattern Recognition 21(6), 581–589 (1988)
13. Wuescher, D.M., Boyer, K.L.: Robust contour decomposition using a constant curvature criterion. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(1), 41–51 (1991)
14. Cronin, T.M.: Visualizing concave and convex partitioning of 2D contours. Pattern Recognition Letters 24(1-3), 429–443 (2003)
15. Mokhtarian, F., Mackworth, A.K.: A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves. IEEE Trans. Pattern Analysis and Machine Intelligence 14(8), 789–805 (1995)
16. Abbasi, S., Mokhtarian, F., Kittler, J.V.: Enhancing CSS-based shape retrieval for objects with shallow concavities. Image and Vision Computing 18(3), 199–211 (2000)
17. Green, P.J.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 82, 711–732 (1995)
18. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian Data Analysis. Chapman and Hall, London (1995)
19. Feder, J.: Fractals. Plenum Press, New York (1988)
20. Valdes, D., Bhalerao, A.: Local Shape Modelling for Brain Morphometry using Curvature Scale Space. In: McKenna, S., Hoey, J. (eds.) Proceedings of the 12th Annual Conference on Medical Image Understanding and Analysis 2008, British Machine Vision Association, pp. 64–68 (July 2008)

Impulse Denoising Using Fuzzy and Directional Processing on a DSP

Alberto Rosales, Volodymyr Ponomaryov, and Francisco Gallegos

Mechanical and Electrical Engineering Higher School,
U.P. Culhuacan Av. Santa Ana, 1000, Col. San Fco. Culhuacan
National Polytechnic Institute of Mexico
{arosaless,vponomar}@ipn.mx

Abstract. This article presents novel fuzzy and directional techniques in denoising colour images contaminated by impulsive random noise. Novel approach has demonstrated excellent performance in image denoising and feature image preservation. Real-Time analysis is realized on a Digital Signal Processor (Texas Instruments).

Keywords: Fuzzy, Directional, Colour, Real-Time, Chromaticity.

1 Introduction

Numerous algorithms have been designed to suppress impulsive noise [1-7]. Here, we present the *Fuzzy Two Step Colour Filter* based on fuzzy and directional processing techniques in denoising colour images, preserving edges, fine details and chromaticity properties. It is improved the performance of mentioned techniques matching in some way their operation to produce a better filtered image.

An interesting fuzzy scheme to remove fixed impulsive noise has been proposed by Schulte [1]; it uses colour information to detect and remove noise, preserving the edges and details in colour images working only in the fixed value outliers not in a random noise, which will be treated in this paper. Other implemented algorithms used here as comparative ones are: AMNF and AMNF2 [2], AMNF3 [3], GVDF [4], where directional techniques are applied efficiently. Also we investigate algorithms: WVDF1, WVDF2, LWVDF, CWVDF and SWVDF [5], they take advantage of adaptive stack filters' design and weighted median filtering framework. Also, PVMM [6], and VMF_SAR [7], which are based in order statistics techniques, are used as comparative ones.

Detailed description of the mathematical background for proposal filter is discussed in Section 2. Impulsive denoising scheme is presented in Section 3; experimental results and criteria used are exposed in Section 4. Finally, conclusions are drawn in Section 5.

2 Fuzzy Two Step Colour Filter (FTSCF)

The colour image pixel in position i is denoted as a vector $\mathbf{x}_i$, which implies the components Red (R), Green (G) and Blue (B), so $\mathbf{x}_i = (x_i^R, x_i^G, x_i^B)$, central

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 554–561, 2008.

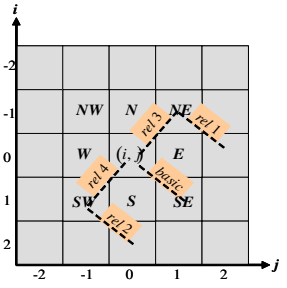


Fig. 1. Basic and Related directions involved for gradient and vectorial calculations

pixel is defined as $\mathbf{x}_c = (x_c^R, x_c^G, x_c^B)$. We use 3×3 processing window in the center of a bigger 5×5 window (Fig. 1).

The procedure consists of calculations in different directions of x_c that correspond to one of the eight directions N=North, E=East, S=South, W=West, NW=North-West, NE=North-East, SE=South-East, SW=South-West (Fig. 1).

The "*basic*" gradient value illustrated in Fig. 1 is defined as: $\nabla x_k^R = \mid x_c^R - x_k^R \mid$, $\nabla x_k^G = \mid x_c^G - x_k^G \mid$, and $\nabla x_k^B = \mid x_c^B - x_k^B \mid$ for each channel, subindex k denotes each neighbors of x_c represented by (i, j) coordinate. We compute the *basic* gradient values for all directions using Fig. 1 and the expression:

$$\nabla(x_c^\beta(i,j), x^\beta(i+k, j+l)) = \mid x_c^\beta(i,j) - x^\beta(i+k, j+l) \mid = \nabla^\beta_{(k,l)} x(i,j), \quad (1)$$

where $k, l \in \{-1, 0, 1\}$, $(i, j) = (0, 0)$, $(k, l) \neq (i, j)$, and β represents the R, G and B components. The eight gradient values associated with the different directions are denoted as "*basic* gradient values". To prevent smoothing, it is proposed to use four *related* gradient values too (Fig. 1), denoted as "*rel*".

2.1 Gradient Values

In Fig. 1 are exposed four *rel* gradient values only for the "*SE*" direction, and their mathematical expressions join to *basic* value are described as follows: for basic gradient value, $\nabla^\beta_{(1,1)} x(i,j) = \nabla^\beta_{SE}$, and for four related gradient values, $\nabla^\beta_{(0,2)} x(i-1, j+1) = \nabla^\beta_{SE(rel1)}$, $\nabla^\beta_{(2,0)} x(i+1, j-1) = \nabla^\beta_{SE(rel2)}$, $\nabla^\beta_{(i,j)} x(i-1, j+1) = \nabla^\beta_{SE(rel3)}$, and $\nabla^\beta_{(i,j)} x(i+1, j-1) = \nabla^\beta_{SE(rel4)}$. We use variable $\gamma = \{\text{N,E,S,W,NW,NE,SE,SW}\}$ to represent each one of the eight neighbor directions.

We refer ∇^β_γ as five gradient values in each direction. Let define two fuzzy sets, SMALL fuzzy set characterizes membership level for a gradient value where if gradient value is sufficiently small, this infers in a big membership value. If membership level is ≈ 1, this implies that *no movement* and *noise* is present in the sample. Opposite case is if the gradient value presents a big difference

among components, that means a big membership level (≈ 1) in BIG fuzzy set. A Gaussian membership function is used for membership values:

$$\mu_{\nabla^{\beta}_{\gamma}(SMALL,BIG)} = \begin{cases} 1, & if(\nabla^{\beta}_{\gamma} < med2, \nabla^{\beta}_{\gamma} > med1) \\ \left(e^{-\left\{\frac{(\nabla^{\beta}_{\gamma}-med2)^2}{2\sigma_1^2}\right\}}, e^{-\left\{\frac{(\nabla^{\beta}_{\gamma}-med1)^2}{2\sigma_1^2}\right\}}\right), & otherwise \end{cases}, \quad (2)$$

where values $med1 = 60, med2 = 10$ and $\sigma_1^2 = 1000$ were obtained experimentally.

2.2 Vectorial Values

Angle among two vectors is computed using expression: $\theta = \cos^{-1}\left\{\frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\|\mathbf{x}_1\| \cdot \|\mathbf{x}_2\|}\right\}$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ represent two pixels with components $\mathbf{x} = (x^R, x^G, x^B)$. It is required to find the angle deviation of a single component in independent way. So, two of three components are omitted from the calculus. This is achieved fixing the value of two components. The next equation presents results for R component, the same equation can be applied for G and B components, also:

$$\theta(x_c^R(i,j), x^R(i+k, j+l)) = \cos^{-1}\left\{\frac{\xi + x^R(i+k,j+l) \cdot x_c^R(i,j)}{\sqrt{\xi + x^R(i+k,j+l)^2} \cdot \sqrt{\xi + x_c^R(i,j)^2}}\right\}, \quad (3)$$

where $\xi = 2 \cdot (255)^2$.

For "SE" direction described in Fig. 1, the *basic* angular deviation value remains as $\theta^R_{(1,1)}x(i,j) = \theta^R_{SE}$, and four values of the related angular deviation are defined as: $\theta^R_{(0,2)}x(i-1,j+1) = \theta^R_{SE(rel1)}$, $\theta^R_{(2,0)}x(i+1,j-1) = \theta^R_{SE(rel2)}$, $\theta^R_{(i,j)}x(i-1,j+1) = \theta^R_{SE(rel3)}$, and $\theta^R_{(i,j)}x(i+1,j-1) = \theta^R_{SE(rel4)}$. To compute "fuzzy angular deviation values" a Gaussian membership functions is used:

$$\mu_{\theta^{\beta}_{\gamma}(SMALL,BIG)} = \begin{cases} 1, & if(\theta^{\beta}_{\gamma} < med3, \theta^{\beta}_{\gamma} > med4) \\ \left(e^{-\left\{\frac{(\theta^{\beta}_{\gamma}-med3)^2}{2\sigma_2^2}\right\}}, e^{-\left\{\frac{(\theta^{\beta}_{\gamma}-med4)^2}{2\sigma_2^2}\right\}}\right), & otherwise \end{cases}, \quad (4)$$

where values $med3 = 0.8, med4 = 0.1$ and $\sigma_2^2 = 0.8$ were obtained experimentally under the best PSNR and MAE results.

2.3 Fuzzy Rules

Proposed fuzzy rules were designed to estimate corruption level in the sample.

Fuzzy Rule 1. It defines the membership level of $x_c^{\beta}(i,j)$ in BIG fuzzy set for each γ direction denoted by $\nabla^{\beta F}_{\gamma}\theta^{\beta F}_{\gamma}$: **IF** $\left(\nabla^{\beta}_{\gamma}\right.$ is BIG AND $\nabla^{\beta}_{\gamma(rel1)}$ is SMALL AND $\nabla^{\beta}_{\gamma(rel2)}$ is SMALL AND $\nabla^{\beta}_{\gamma(rel3)}$ is BIG AND $\left.\nabla^{\beta}_{\gamma(rel4)}\right.$ is BIG$\left.\right)$ AND $\left(\theta^{\beta}_{\gamma}\right.$ is BIG AND $\theta^{\beta}_{\gamma(rel1)}$ is SMALL AND $\theta^{\beta}_{\gamma(rel2)}$ is SMALL AND $\theta^{\beta}_{\gamma(rel3)}$ is BIG AND $\left.\theta^{\beta}_{\gamma(rel4)}\right.$ is BIG$\left.\right)$ **THEN** $\nabla^{\beta F}_{\gamma}\theta^{\beta F}_{\gamma}$ is BIG. AND operator outside of the parenthesis in fuzzy rule is defined as $\min(A,B)$, and inside the parenthesis this is defined as A AND $B = A * B$.

A colour component is considered *noisy* or that belongs to BIG fuzzy set if its basic values $\mu_{\nabla_\gamma^\beta} \approx \mu_{\nabla^\beta_{\gamma(rel3)}} \approx \mu_{\nabla^\beta_{\gamma(rel4)}}$,$\mu_{\nabla_\gamma^\beta} \neq \mu_{\nabla^\beta_{\gamma(rel1)}} \neq \mu_{\nabla^\beta_{\gamma(rel2)}}$,$\mu_{\theta_\gamma^\beta} \approx \mu_{\theta^\beta_{\gamma(rel3)}} \approx \mu_{\theta^\beta_{\gamma(rel4)}}$, and $\mu_{\theta_\gamma^\beta} \neq \mu_{\theta^\beta_{\gamma(rel1)}} \neq \mu_{\theta^\beta_{\gamma(rel2)}}$. In this way, it is bonded in some level if central pixel is a noisy one and with local movement. Indiscrimining among noise and movement levels, it was designed the following fuzzy rule:

Fuzzy Rule 2, defines a fuzzy noise factor denoted as r^β:

IF $\nabla_N^{\beta F}\theta_N^{\beta F}$ is BIG OR $\cdots$ OR$\nabla_{SE}^{\beta F}\theta_{SE}^{\beta F}$is BIG **THEN** $r^\beta$is BIG, where $N, SE \in \gamma$.

Fuzzy noise factor is used in the algorithm as a threshold to determine if central pixel is corrupted or belongs to an edge or a fine detail (local movements). Fuzzy rule 2 contains seven disjunctions, which are represented by triangular co-norms defined as: $\max(A, B)$.

3 Impulsive Random Denoising

The following condition must be satisfied to process a noisy central pixel instead of edge or fine detail:

`If` $r^\beta \geq 0.3$ `then the filtering is realized using fuzzy membership levels obtained by the` BIG `fuzzy set, in other case the filter's output is` $y^\beta_{output} = x^\beta_c$.

Threshold value of 0.3 was obtained experimentally in agree to the best results for different criteria PSNR, MAE, and NCD. If $r^\beta \geq 0.3$, we take the fuzzy values as the weights for each a neighbor component. To provide the weights we propose to use a standard negator operator described as $\zeta_s(x) = 1 - x$ with $x \in [0, 1]$ to derive membership level in "NO BIG" fuzzy set (noise free). Final expression is $\zeta_\gamma^{\beta F} = 1 - \nabla_\gamma^{\beta F}\theta_\gamma^{\beta F}$ representing fuzzy weights used for the filtering algorithm. The weight value for the central component in NO BIG fuzzy set is $\zeta_c^{\beta F} = 3 \cdot \sqrt{1 - r^\beta}$.

Now, magnitude component values are ordered in agree to the following procedure:

$$x_\gamma^\beta = \{x_{SW}^\beta, x_S^\beta, x_{SE}^\beta, x_W^\beta, x_c^\beta, x_E^\beta, x_{NW}^\beta, x_N^\beta, x_{NE}^\beta\},$$
$$x_\gamma^{\beta(1)} \leq x_\gamma^{\beta(2)} \leq x_\gamma^{\beta(3)} \leq x_\gamma^{\beta(4)} \leq x_\gamma^{\beta(5)} \leq x_\gamma^{\beta(6)} \leq x_\gamma^{\beta(7)} \leq x_\gamma^{\beta(8)} \leq x_\gamma^{\beta(9)} \underset{implies}{\Longrightarrow}$$
$$\zeta_\gamma^{\beta(1)} \leq \zeta_\gamma^{\beta(2)} \leq \zeta_\gamma^{\beta(3)} \leq \zeta_\gamma^{\beta(4)} \leq \zeta_\gamma^{\beta(5)} \leq \zeta_\gamma^{\beta(6)} \leq \zeta_\gamma^{\beta(7)} \leq \zeta_\gamma^{\beta(8)} \leq \zeta_\gamma^{\beta(9)}$$

Filtering procedure is performed selecting one of the neighbor components, this avoids smoothing of the details and edges. Ordering properties outperforms the filtering results, withdrawing pixels farther respect to the central one. The complete filtering procedure is developed employing the next methodology: let define a variable $sum^\beta += \zeta_\gamma^{\beta(j)}$, with j decreasing from 9 to 1. Decreasing of j

is valid until $sum^{\beta} \geq \zeta^{\beta}$ to be satisfied, where $\zeta^{\beta} = \left(\Sigma_{\gamma}\zeta_{\gamma}^{\beta F} + 3 \cdot \sqrt{1 - r^{\beta}}\right) \Big/ 2$ is the total weight to be considered. If this condition is fulfilled, the decreased value of j must not be so small due to ordered set values, where j could be assigning the magnitude values that are probably corrupted. To solve this, it is proposed the following condition: if $j \leq 2$, total weight ζ^{β} is upgraded using $\zeta^{\beta} = (\zeta^{\beta} - (\zeta_{\gamma}^{\beta(1)} + \zeta_{\gamma}^{\beta(2)})) \Big/ 2$, and we define a variable $sum'^{\beta} += \zeta_{\gamma}^{\beta(j)}$ with j decreasing from 9 to 3; decreasing of j is valid until $sum'^{\beta} > \zeta^{\beta}$. At moment when j satisfies this condition, it is selected the jth ordered value as the output's filtered value, or if condition $j \leq 2$ was not covered, it is selected the jth ordered value as the output's filtered value, so: $y_{output}^{\beta} = x_{\gamma}^{\beta(j)}$.

4 Experimental Results

Standard criteria: Pick Signal Noise Relation (PSNR)[3], Mean Absolute Error (MAE) [8], and Normalized Colour Difference (NCD) [9], were used to characterize the performance of proposed technique.

Under experimental tests, we found the optimum parameter values presenting the best numerical results in criteria by the novel filter FTSCF. Colour test images (Lena, Baboon, Peppers, Parrots and Goldhill) in 24-bits presentation, a true colour RGB space, 320×320 pixels were used. Images were contaminated by impulsive random noise. This noise was applied in each a channel of the image in independent way. Other algorithms implemented as comparative ones were (according to references): ABST_* and Wilc_* filters family [3], MAMNFE, AMNF, AMNF2 [10], AMNF3 [3], AVMF filter [11], SAP*-* and FRWA*-* filters family [12], BVDF and GVDF*-* filters family [4], INR [1], WVDF1, WVDF2, LWVDF, CWVDF and SWVDF filters [5], MMKNN*-* filters family [3] [13], and finally, PVMM algorithm [6].

In Table 1, one can see that the best results are presented by novel filter in low and middle noise levels (from 0% to 30%) giving the best chromaticity properties preservation.

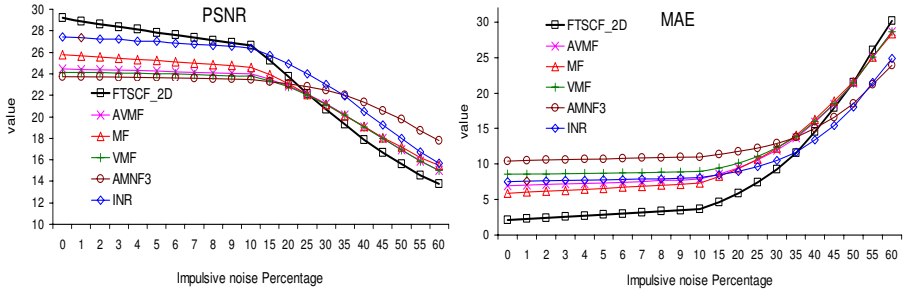
In Fig. 2, the results for PSNR and MAE criteria for Baboon colour image are exposed. FTSCF works efficiently under PSNR criterion from 0% to 10% of corruption, after that level the algorithm falls in its performance, presenting second best results until 20% of corruption. In MAE criterion, the best results are obtained by proposed filter from 0% to 35% of noise corruption.

In Fig. 3, for Pepper image, FTSCF works fine in agree to PSNR criterion in range of corruption from 0% to 10%. For FTSCF in Fig. 4, we appreciate a high quantity of noisy pixels than those observed applying INR filter, but in details preservation as numerical results indicates, novel filter is the best algorithm.

Analyzing Real-Time processing we use a Digital Signal Processor manufactured by Texas Instruments (TI). This device is the Evaluation Module DM642. It has video port intefaces, codecs, memory, and software tools to implement

Table 1. NCD results for Lena colour image

	NCD CRITERION									
Impulsive Noise Level	0	5	10	15	20	25	30	40	50	60
ABST_CMS	0,0195	0,0203	0,0213	0,0226	0,024	0,0258	0,0283	0,0359	0,0465	0,063
AMNF	0,0199	0,0209	0,0225	0,0249	0,0281	0,0325	0,0376	0,0514	0,067	0,084
AMNF2	0,019	0,0199	0,0207	0,0217	0,0228	0,0244	0,0264	0,0338	0,0453	0,062
AMNF3 (AMNF2_CMS)	0,0188	0,0195	0,0203	0,0212	0,0222	0,0234	0,0252	0,0316	0,0418	0,058
AVMF	0,0078	0,0096	0,0116	0,014	0,0165	0,0196	0,0235	0,0341	0,0483	0,068
FATM_p2	0,0178	0,0187	0,0201	0,0222	0,0252	0,0292	0,034	0,0472	0,0622	0,08
MF	0,0137	0,0149	0,0163	0,0181	0,02	0,0228	0,0261	0,0365	0,0503	0,068
FRWA2_2	0,0189	0,0204	0,0223	0,0244	0,0276	0,0312	0,0362	0,0496	0,0665	0,086
FTSCF	**0,0017**	**0,0037**	**0,0059**	**0,0084**	**0,0115**	**0,0158**	**0,0206**	0,0352	0,0537	0,0771
VMF	0,0154	0,0162	0,0171	0,0185	0,02	0,0221	0,0252	0,0348	0,0485	0,068
BVDF	0,0169	0,0185	0,0207	0,0234	0,0265	0,0307	0,0363	0,0525	0,0745	0,101
GVDF_DW	0,0183	0,0202	0,0224	0,0246	0,0262	0,0282	0,0305	0,0378	0,0504	0,071
GVDF	0,0162	0,0175	0,0187	0,0199	0,0214	0,0232	0,0268	0,038	0,0547	0,076
GVDFAD	0,0154	0,0162	0,0173	0,0186	0,0201	0,0221	0,0249	0,0337	0,0462	0,064
INR	0,0164	0,0166	0,0171	0,0176	0,0184	0,0197	0,0214	**0,0281**	**0,0405**	**0,059**
CWVDF	0,0092	0,0112	0,0138	0,0174	0,0222	0,0282	0,0357	0,0552	0,0782	0,104
SWVDF	0,0046	0,0095	0,0144	0,0193	0,024	0,0291	0,0353	0,052	0,0742	0,101
LWVDF	0,017	0,0187	0,0208	0,0234	0,0265	0,0307	0,0362	0,0524	0,0744	0,102
MAMNFE	0,0191	0,0202	0,0213	0,023	0,0248	0,0275	0,0309	0,0412	0,0552	0,072
MMKNN_CMS	0,0159	0,0168	0,0178	0,0191	0,0206	0,0225	0,0251	0,0333	0,0452	0,063
PVMM	0,0117	0,016	0,0221	0,0285	0,0361	0,0445	0,0524	0,0708	0,0882	0,106
SAP-2_2	0,0197	0,0208	0,0223	0,0241	0,0262	0,0287	0,0319	0,042	0,0556	0,073
SAP-AD2_1	0,0122	0,0158	0,0194	0,0229	0,0272	0,0319	0,0369	0,0496	0,0651	0,083
Wilc_CMS	0,0073	0,0096	0,0128	0,0166	0,0207	0,0255	0,0302	0,0434	0,0592	0,079
WVDF1	0,0137	0,0156	0,0181	0,0213	0,025	0,0302	0,0367	0,0544	0,0766	0,103
WVDF2	0,0113	0,0136	0,0165	0,0205	0,0251	0,0313	0,0385	0,0575	0,0798	0,105



a) b)

Fig. 2. Baboon image filtering results for different criteria

the designs in an efficient way. We are basing our proposals under the "Reference Structure 5 (RF5)" [14] designed by TI. RF5 is useful in working on video processing due to its designed structure to manage a lot of simultaneous signals (R,G and B channels) and processing them using different algorithms to provide the final result. The RGB format managed here is 565, those means, 5-bits to Red, 6-bits to Green and 5-bits to Blue channels. This produces in Red and Blue channels, 32 colour levels, and in Green channel, 64 colour levels. Video was processed in QCIF format (176×144 pixels). Time processing values needed to realize some algorithms are given in Table 2, where it is noted that our algorithm works fine against GVDF, SAP-2_2, and SAP-AD2_1. Last algorithms includes complex operations. Time processing for FM and VMF algorithms are presented to compare common filters and robust ones.

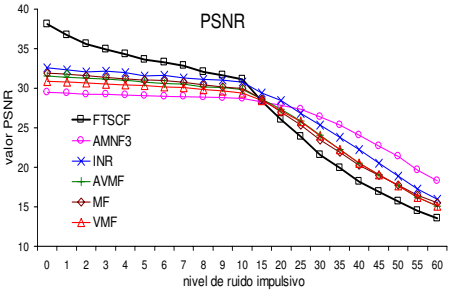


Fig. 3. Peppers image filtering results for PSNR criterion

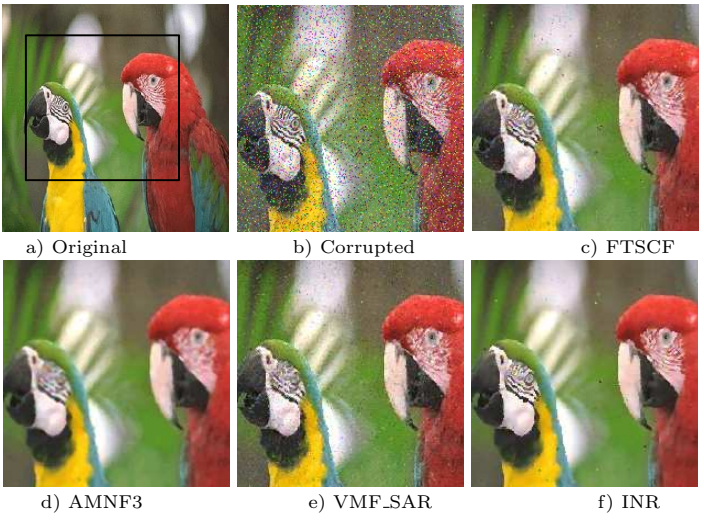


Fig. 4. Zooming of Parrots image filtering with 20% of impulsive random noise

Table 2. Real-Time performance for 20 video colour frames processing

Algorithm	Total Time (secs.)	Max. Time (secs.)	Average Time (secs.)
MF	0.062	0.0037	0.0031
VMF	0.552	0.0283	0.0278
FTSCF	**108.783**	**5.442**	**5.439**
AMNF3	85.917	7.216	4.295
GVDF	117.382	5.887	5.869
SAP-2_2	174.741	8.816	8.737
SAP-AD2_1	232.504	11.729	11.625

5 Conclusions

Proposed FTSCF presents good performance according to different criteria PSNR, MAE, and NCD in noise suppression, detail preservation and chromaticity properties, respectively. FTSCF permits connecting two powerful techniques

in image processing. In directional processing techniques, the pixels are taken as vectors and angular deviations are obtained. On the other hand, fuzzy logic procedures are used to compute membership values. FTSCF's time processing presents adequate performance in comparison with other robust algorithms.

Acknowledgments. Authors would thank CONACYT and National Polytechnic Institute of Mexico for their support to realize this work.

References

1. Schulte, S., Morillas, S., Gregori, V., Kerre, E.: A New Fuzzy Color Correlated Impulse Noise Reduction Method. Trans. on Image Proc. 16(10), 2565–2575 (2007)
2. Plataniotis, K.N., Androutsos, D., Vinayagamoorthy, S., Venetsanopoulos, A.N.: Color Image Processing Using Adaptive Multichannel Filters. IEEE Transactions on Image Processing 6(7), 933–949 (1997)
3. Ponomaryov, V.I., Gallegos-Funes, F.J., Rosales-Silva, A.: Real-time color imaging based on RM-Filters for impulsive noise reduction. Journal of Imaging Science and Technology 49(3), 205–219 (2005)
4. Trahanias, P.E., Venetsanopoulos, A.N.: Vector Directional Filters. A new class of multichannel image processing Filters. IEEE. Trans. on Image Processing 2, 528–534 (1993)
5. Lukac, R., Smolka, B., Plataniotis, K.N., Venetsanopoulos, A.N.: Selection Weighted Vector Directional Filters. Comput. Vision and Image Unders. 94, 140–167 (2004)
6. Alajlan, N., Kamel, M., Jernigan, E.: Detail Preserving Impulsive Noise Removal. Signal Processing: Image Communication 19(10), 993–1003 (2004)
7. Smolka, B., Lukac, R., Chydzinski, A., Plataniotis, K.N., Wojciechowski, W.: Fast Adaptive Similarity Based Impulsive Noise Reduction Filter. Real-Time Imaging 9(4), 261–276 (2003)
8. Bovik, A.: Handbook of Image and Video Processing. Academic Press, San Diego (2000)
9. Poynton, C.A.: Poynton's Color FAQ, Electronic Preprint (1995)
10. Plataniotis, K.N., Androutsos, D., Vinayagamoorthy, S., Venetsanopoulos, A.N.: Color Image Processing Using Adaptive Multichannel Filters. IEEE Transactions on Image Processing 6(7), 933–949 (1997)
11. Lukac, R.: Adaptive Vector Median Filtering. Pattern Recognition Letters 24, 1889–1899 (2003)
12. Szczepanski, M., Smolka, B., Plataniotis, K.N., Venetsanopoulos, A.N.: On the Geodesic Paths Approach to Color Image Filtering. Signal Processing 83, 1309–1342 (2003)
13. Ponomaryov, V.I.: Real-Time 2D-3D filtering using order statistics based algorithms. Journal of Real-Time Image Processing 1(3), 173–194 (2007)
14. Reference Frameworks for eXpressDSP Software: RF5, An Extensive, High Density System (SPRA795A)

A Motion Planning Strategy for Rapidly Finding an Object with a Mobile Manipulator in 3-D Environments

Alejandro Sarmiento[1], Judith Espinoza[2], Rafael Murrieta-Cid[2], and Seth Hutchinson[1]

[1] University of Illinois, Beckman Institute
Urbana, Illinois
{asarmien,seth}@uiuc.edu
[2] Centro de Investigación en Matemáticas, CIMAT
Guanajuato, México
{jespinoza,murrieta}@cimat.mx

Abstract. In this paper, we address the problem of searching for an object in a 3-D environment. We consider a mobile manipulator with an "eye-in-hand" sensor moving in the 3-D environment. In particular, we consider a static object whose location is modeled with a probability density function (pdf). We generate routes that minimize the expected value of the time until the object is first seen when following the route. We use a sample-based convex cover to estimate the size and shape of visibility regions in 3-D. The resulting convex regions are exploited to generate trajectories that compromise between moving the manipulator base and moving the robotic arm.

1 Introduction

This work addresses the problem of finding a static object. In this paper, our goal is for the robot to find the object as quickly as possible on average. We claim that our work presents a new paradigm for search tasks, where it is important to gain as much new information in the shortest time as possible. This can be very useful in applications where the time assigned to the task is limited or not completely known. We present a discrete formulation, in which we use a visibility-based decomposition of the environment to convert the problem into a combinatoric one.

The possible applications have a wide range, from finding a specific piece of art in a museum to search and detection of injured people inside a building. This problem is closely related to the coverage problem [4] in the sense that any complete strategy to find an object must sense the whole environment.

In [1] we have investigated the problem of finding an object in a 3D environment *for the case of a point robot*. In that work, we have introduced a probabilistic sampling method to decompose the workspace into convex regions. In this paper we use that convex region partition. The research reported in this paper differs from our previous efforts in the following main points:

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 562–572, 2008.

1. We consider a mobile manipulator with an "eye-in-hand" sensor.
2. We investigate the use of different metrics to quantify the paths cost.
3. We use complete regions as sensing locations as opposed to a single location (point).

Note that current technology make feasible the assumption of equipping a mobile manipulator with an "eye-in-hand" sensor. For instance, using the very small omnidirectional camera described in [3]. Below, we present the definition of our problem.

2 Problem Definition

This work addresses the problem of minimizing the expected value of the time to find an object in a known 3-D workspace.

In general terms, we define the problem of searching for an object as follows: Given one mobile manipulator robot with sensing capabilities, a completely known 3D environment and an object somewhere in the world, develop a motion strategy for the robots to find the object in the least amount of time on average.

The environment W is known, and modeled as a set of polyhedrons. The obstacles generate *both motion and visibility constraints*. Furthermore, we assume that the probability of the object being in any specific point is uniformly distributed. Therefore, the probability of the object being in any subset $C_i \subset W$ is proportional to the volume of C_i. Where C_i is a convex region in a 3-D workspace.

The robot senses the environment at a set of locations L_i (also known as *guards*, from the art gallery problem [11]). The visibility region of location L_i is considered to be the volume of the convex region denoted C_i. Note that all points inside C_i can be connected by a clear line of sight from any location (point) L_i inside C_i. Note also that, it gives flexibility as to where to place the sensor, any point L_i inside C_i is a valid candidate.

The set $\{C\}$ is chosen so that the union of all C_i covers the whole environment, that is, $\bigcup_i C_i = W$. We do not require nor assume the set $\{C\}$ to be minimal.

Our exploration protocol is as follows: the robot always starts at a particular region in $\{C\}$ (associated to the starting point) and visits the other regions as time progresses. It follows the shortest paths for some given metric between them. We also investigate the use of different metrics to quantify the paths cost. We generate, metric-dependent trajectories, that is, trajectories that find a compromise between moving the base or moving the robotic arm.

The robot only gathers information about the environment (sensing) when it reaches one different convex region – it does not sense while moving. We describe the route followed by the robot as a series of convex regions C_{i_k} that starts with the robot's initial region and includes every other region at least once. Note that while C_i refers to region in the environment, C_{i_k} refers to the *order* in which those regions are visited. That is, the robot always starts at C_{i_0}, and the k-th region it visits is referred to as C_{i_k}.

For any route R, we define the time to find the object T as the time it takes to go through the regions – in order – until the object is first seen.

Our goal is to find the route that minimizes the expected value of the time it takes to find the object

$$E\left[T|R\right] = \sum_{j} t_j P\left(T = t_j\right) \tag{1}$$

where

$$P\left(T = t_j\right) = \frac{Volume\left(C_{i_j} \setminus \bigcup_{k<j} C_{i_k}\right)}{Volume(W)}. \tag{2}$$

Here, t_j is the time it takes the robot to go from its initial position – through all regions along the route – until it reaches the j-th *visited* region C_{i_j}, and $P\left(T = t_j\right)$ is the probability of seeing the object for the first time from region C_{i_j}. Since the robot only senses at specific regions, we also denote this probability of seeing the object for the first time from region C_{i_j} as $P\left(C_{i_j}\right)$.

2.1 Expected Value vs. Worst Case

It is important to note the difference between minimizing the expected value of the time to find an object and minimizing the time it would take in the worst case.

To minimize the worst case time, the robot must find the shortest path (for instance, in euclidean sense) that completely covers the environment (the shortest watchman tour problem [5]). This usually means that no portions of the environment are given any priority over others and the rate at which new portions of the environment are seen is not important.

On the other hand, to minimize the expected value of the time, the robot must gain probability mass of seeing the object as quickly as possible. For a uniform probability density function (pdf), this translates into sensing large portions of the environment as soon as possible, even if this means spending more time later to complete covering the whole environment. We believe this represents another paradigm for search tasks, where it is important to gain as much new information in the shortest time possible.

The trajectories that satisfy the previous two criteria are not the same. In fact, for a given environment, the route that minimizes the distance traveled may not minimize the expected value of the time to find an object along it.

Consider the example in Fig. 1. The robot starts in the corridor at location L_0. Assume that the object will always be in one of two rooms, and the probability of it being in either is related to the size of the room. These rooms have a narrow door and the entire room is visible from the threshold. The room to the right – seen from location L_1 – is smaller but lies closer to the initial location, while the room to the left – seen from L_2 – is larger but farther from initial position. There are only two routes the robot might take to solve this problem: Go to the smaller room first ($L_{0_0} \rightarrow L_{1_1} \rightarrow L_{2_2}$), or go to the larger room first ($L_{0_0} \rightarrow L_{2_1} \rightarrow L_{1_2}$). For the following analysis, we assume that the robot moves at a constant speed of 1 unit per second.

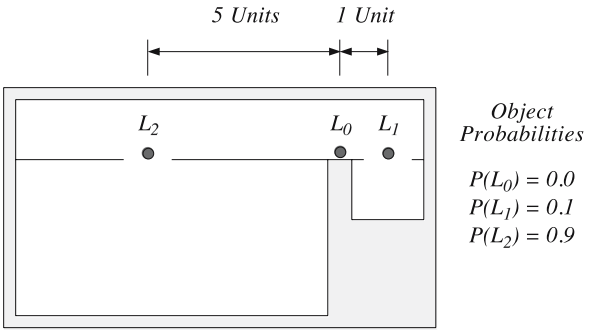


Fig. 1. Example with a simple environment

Route 1 – If the robot goes to the smaller room first and then moves on to the larger room, it reaches L_1 at time 1 and L_2 at time 7. The expected value of the time it takes to find the object following this route is

$$E\left[T\right|\left(L_0, L_1, L_2\right)] = (0.1)(1) + (0.9)(7) = 6.4.$$

The robot always completes its search after 7 seconds.

Route 2 – If the robot moves to the larger room first and then goes to the smaller room, it reaches L_2 at time 5 and L_1 at time 11. The expected time in this case is

$$E\left[T\right|\left(L_0, L_2, L_1\right)] = (0.9)(5) + (0.1)(11) = 5.6.$$

In the worst case, it will take the robot 11 seconds to find the object.

A robot following route 1 always finishes searching after 7 seconds, while a robot following route 2 takes 11 seconds. Route 1 minimizes the distance traveled. However, the average time it takes for a robot following route 1 to find the object is 6.4 seconds whereas for route 2 it is only 5.6 seconds. Route 2 minimizes the expected value of the time to find an object.

Thus, *a trajectory that is optimal in the distance traveled does not necessarily minimize the expected value of the time to find an object along it.*

3 Approach Overview

This work builds on our previous research on expected value search with mobile robots. In [2] we proposed an approach to solve this problem in a 2-D polygonal environment. The basic idea was to define a set of sensing locations from which the whole environment is covered. We then defined a graph over these locations. We shown that even the problem of finding an object in a 2-D polygonal workspace with a point robot is NP-hard. Therefore, we propose the heuristic of a utility function defined as the ratio of a gain over a cost. We use this utility function to drive a greedy algorithm in a reduced search space that is able to

explore several steps ahead without incurring too high a computational cost. The present work builds on the same problem but in a three dimensional workspace with a mobile manipulator robot.

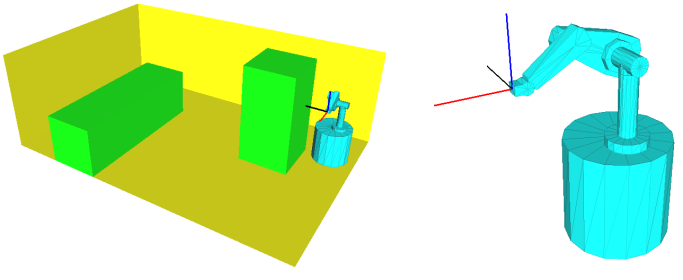
The expected value of the time depends on two factors: 1) The distance traveled (equivalent to the time if the robot moves to constant saturated speed), quantified in some given metric, which represents the *cost* and 2) the probability mass of seeing the object, which is equivalent to the *gain*.

We provide algorithms to compute the shortest paths (for a given metric) for a 7 degrees of freedom mobile manipulator to move between configurations, –cost. We also provide visibility-based decompositions of the environment to compute the probability mass of seeing the object, –gain. We use a set of sensing locations. Then we link them in a graph and perform a graph search to generate the global trajectory.

We propose a sample-based convex cover algorithm that allows us to partition the environment into convex pieces and also to estimate the volume of each piece. Using our convex cover decomposition, we propose an approach to our search problem, for the mobile manipulator with an "eye-in-hand" sensor like the one shown in Fig. 2, moving in a 3-D environment.

We use the complete regions as sensing locations, that is, as long as the end effector (which is equipped with a sensor) is inside one of them, we know that the vast majority of the whole region will be visible to the robot. This gives flexibility as to where to place the sensor, and is also helpful in the generation of what we call "metric-dependent trajectories." These trajectories take into account weights assigned to the different degrees of freedom and therefore, can be good compromises between moving the base or moving the robotic arm.

Finally, we use our previously proposed greedy approach [2] to search the resulting graph and generate trajectories that minimize the expected value of the time to find an object in the environment. Unlike our previous work, the proposed algorithms are no longer complete nor deterministic, but we do provide probabilistic bounds on how good the coverage is.



(a) (b)

Fig. 2. a) 3-D workspace b)A mobile manipulator with an "eye-in-hand" sensor

4 Sample-Based Sensing Strategy in 3-D

Our convex cover and subsequent graph abstraction capture the connectivity of the workspace. There have been many related approaches that connect samples to capture connectivity of high dimensional spaces, for example, [8,7,10,12] just to name a few. Our work is different in several aspects. First, we are interested in representing the workspace for searching an object, not in abstracting the configuration space for path planning purposes. Second, most techniques throw away "useless" samples, we keep all of them since we need to estimate the size of volumes. Finally, we can determine a clear-cut threshold of when it is no longer useful to continue sampling.

Due to the high computational complexity of visibility computation in 3-D [11], we decided to use sampling to decompose the environment into convex regions and to estimate the size of those regions. In particular, we implemented the sampling strategy in [12] originally proposed to construct a probabilistic roadmap in the configuration space. The difference in our case is that we will sample the workspace (not the configuration space) and keep all samples, as opposed to throwing away those that do not join important configurations.

In order to capture the size and shape of the workspace W, we generate a set of independent, uniformly distributed collision free samples S in W. Among these samples, we choose a *hidden guard set* G. A set is called a hidden guard set if it covers the environment and individual member of the set are not visible to each other. Since we are approximating W with sample points, for the set G to cover the environment, every sample in S must be visible to at least one guard in G. That is,

$$\bigcup_{g_i \in G} Vis\,(g_i) = S, \tag{3}$$

where $Vis\,(g_i)$ is the set of points in S whose line segment to g_i does not intersect the workspace W. Also, since guards must not be mutually visible, $g_i \notin Vis\,(g_k) \quad \forall k \neq i$.

To determine how well the workspace has been covered, consider an environment of unit volume like the one shown in Fig. 3. Suppose that the portion of the environment that is visible from the guard set is A and has measure $\mu(A) = 1-\epsilon$, while B is another portion with measure $\mu(B) = \epsilon$ that is not yet visible and does not contain a single sample point.

If samples are drawn independently, the probability of m consecutive points not falling into the uncovered region B is $P\,(A_m) = (1-\epsilon)^m$. After m consecutive samples, it is still possible that the actual size of B is greater than ϵ, but with a large m, we can bound this probability with a small value α. For this, we determine m as follows,

$$(1-\epsilon)^m \leq \alpha; \quad m\log(1-\epsilon) \leq \log(\alpha); \quad m \geq \frac{\log(\alpha)}{\log(1-\epsilon)}\ . \tag{4}$$

Therefore, choosing a large enough m we can expect with certainty $(1-\alpha)$ that the size of the unseen region B is at most ϵ.

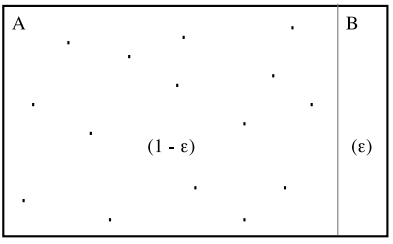


Fig. 3. An environment of unit volume and the sizes of covered (A) and uncovered (B) regions

4.1 Convex Cover Algorithm

Given our strategy of stochastically choosing a hidden guard set, there will be a neighborhood around each guard that only that particular guard can see (since a small perturbation on a guard is not likely to make it visible to the others). Likewise, provided an adequate number of samples, there will be a set of sample points that only one particular guard can see. We call this set of points, the *kernel* of the guard, and denote it as

$$Ker\,(g_i) = Vis\,(g_i) \setminus \bigcup_{k \neq i} Vis\,(g_k)\,. \tag{5}$$

In any minimum convex cover $\{C\}$, each convex region C_i has a set of points only contained in that particular region. Otherwise, region C_i could simply be removed and the cover would not have been minimal. Although we know there is not an exact equivalence, we use guard kernels as an approximation to these unique subsets. Thus, the main idea behind our convex cover algorithm, is that by "growing" convex regions around the guard kernels, we can generate a low cardinality convex cover. More details about our algorithm to generate a convex cover environment partition can be found in [1].

5 Application to a Mobile Manipulator

We have applied our search strategy to a mobile manipulator with an "eye-in-hand" sensor, like the one shown in Fig. 2 b). This robot is made of an arm with four degrees of freedom (DOFs) mounted on a mobile base with three DOFs (translation and rotation in the plane). Since the first rotation of the arm (around a vertical axis) and center of rotation for the mobile base are off-center, they are not the same axis. Thus, the entire system has a total of seven internal DOFs – two translations and five rotations, see figure 2 (b). The figure also shows the coordinate frame for the end effector, where our sensor resides.

The decomposition into regions provides flexibility on exactly where to place the end effector to sense each region – any point inside is a valid candidate. On one hand, this simplifies the path planning problem, since we have a *set* of goal

configurations, as opposed to a single one. But on the other, it invites a more challenging problem: given that we have more options, what is the *best* way to reach one of these regions? We are not only interested in finding a way to reach a region, but also the best way to do it.

We want to find the path, which minimizes the expected value of the time to find an object. But, recall that the computation of the expected value of the time depends on two main factors 1) the distance traveled (cost) and 2) the probability mass of seeing the object (gain). Thus, we also need to find shortest paths for a given metric (cost) between configurations for a 7 degrees of freedom mobile manipulator. The choice of metric is interesting because it can generate different "behaviors" for the robot, as we will explain in the next subsection.

5.1 Metric-Dependent Trajectories

We would like to find shortest paths between configurations. The actual paths depend on the metric used to measure distance. One way to define the distance between two configurations X and Y in a D-dimensional configuration space is

$$\|X - Y\|_{\Lambda} \equiv (X - Y)^T \Lambda (X - Y), \tag{6}$$

where Λ is a diagonal matrix with positive weights $\lambda_1, \lambda_2, \ldots \lambda_D$ assigned to the different DOFs. In general, the matrix Λ is needed because not all joints might be equivalent (e.g., large links vs. small links), and also because different DOFs might not even be measuring the same movement (translation vs. rotation).

In our case, by weighting each DOF differently we can assign different priorities to the two main components of our system: the mobile base and the robotic arm. We divided the DOFs of the mobile manipulator into two groups, the two DOFs that translate the robot on the plane and the five rotations that further place the end effector at a specific position.

If we assign larger weights to one group and smaller weights to the other, it will be reflected in the shortest paths for that metric. It is evident that these paths will have lower length when the DOFs that move the most are the ones with the low cost. Since we can select which DOFs we would like to move the most (or the least), we can find trajectories that find a compromise between, for example, moving the base and moving the arm.

The problem is, of course, to find the shortest path to a set of configurations (the convex region). To find that shortest path, we implemented a wavefront expansion [9]. This algorithm restricted to a *particular* robot has polynomial complexity .

The wavefront expansion algorithm uses the weights $\lambda_1, \lambda_2, \ldots \lambda_D$ assigned to the DOFs to expand the wave in the different dimensions, so that all configurations at the growing boundary are at the same distance from the start configuration.

5.2 Simulation Results

To make the results more evident, we divided the DOFs of the robot into two groups, two translations and five rotations. First we assigned a larger weight

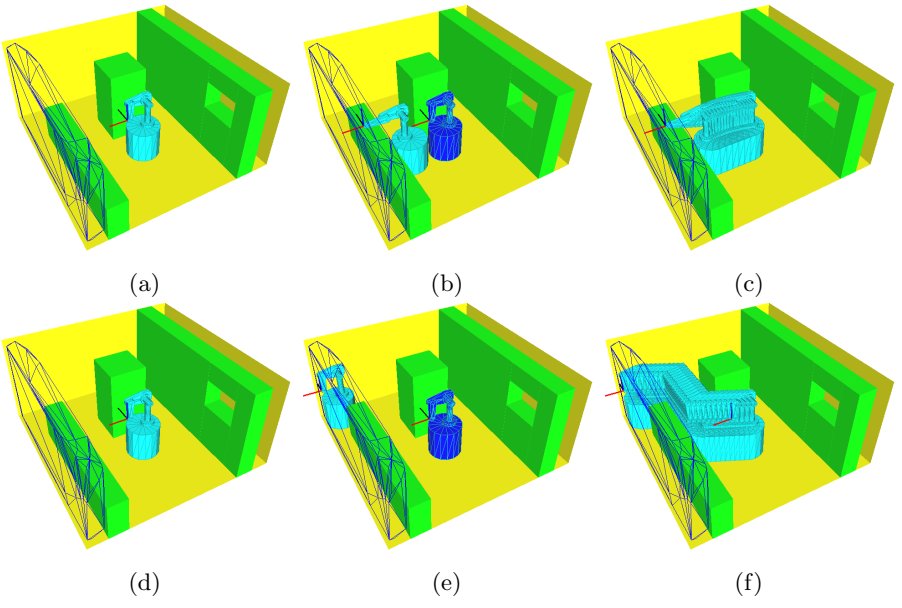


Fig. 4. Visiting one region: Low-Cost Translation vs Low-Cost Rotation

to one group and then to the other. Figure 4 Parts (a) and (d) show the initial position and one region that need to be visited. Parts (b) and (e) show the initial (dark) and final (light) configurations. Finally, parts (c) and (f) show the actual path followed by the robot.

In the example shown in figure 4, the robot only visits one region, but we assign different weights to the rotation and translation DOFs. Parts (a) to (c) show the case when the rotation's DOFs have a small weight, in contrast, parts (d) to (f) show the case when the translation's DOFs have a small weight. Note that, the resulting trajectories are very different. When rotations are better (Fig. 4 (c)), the robot goes straight to the obstacle and places its sensor over it. Note that the robot must rotate the base (and compensate with the "shoulder") and also rotate the middle link (and compensate with the end effector) to be able to reach beyond the obstacle, so there are a total of four rotations in this movement. In contrast, when it is better to translate (Fig. 4 (f)), the robot does not rotate any link and simply moves around the obstacle to reach the region behind.

Figure 5 shows an example where the robot must visit several regions to see the whole environment. Here, the global plan corresponds to the order of visiting the regions, such order minimizes the expected value of the time to find an object, and is computed using our algorithm proposed in [2]. Figure 5, parts (a), (b) and (c) show the case of low-cost rotation. Figures 5 (a) and (b) show snapshots of the robot path and the regions covered through the path. Figure 5 (c) shows the whole path. In this case, the robot chooses to see first the region associated to the hole in the wall. It requires several arm's rotations, then the

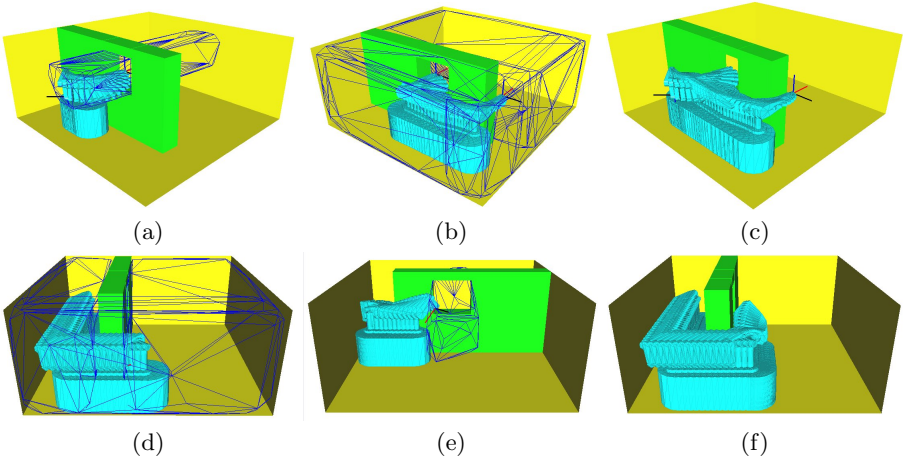


Fig. 5. Visiting several regions: Low-Cost Translation vs Low-Cost Rotation

robot translates as few as possible (because the translation is expensive) to cover the remaining regions. Note that the robot goes, just around the corner to cover the whole environment.

Figure 5, parts (d), (e) and (f) show the case of low-cost translation, the whole path is shown in figure 5 (f). Now, the robot chooses a path which requires a long translation around the wall, the robot see the region associated to the hole in the wall up to the end. In this second case, the robot avoids to stick its arm through a hole in the wall, because it requires several expensive rotations. The expected value of the time depends on both: 1) the volume of the regions and 2) the distance traveled (and hence the metric used to quantify it). Therefore, the order for visiting regions may change according to the metric used to quantified the path cost, thus generating robot's *behaviors*.

6 Conclusions and Future Work

We addressed the problem of searching for an object in a known 3-D environment. Due to the high computational complexity of visibility queries in 3-D, we decided to use a sample-based approach to approximate the size and shape of visibility regions. The decomposition into regions gives us flexibility on exactly where to place the sensor, and their convexity guarantees that the entire cell will be visible from every point inside.

Based on this covering, we presented approaches for a mobile manipulator with an "eye-in-hand" sensor. We have also presented an approach to generate metric-dependent trajectories, that is, trajectories that compromise between moving the base and moving the robotic arm.

There are several extensions to the proposed problems that we think would be interesting to research. In this work, we assumed that the pdf modeling the

object location was uniform, i.e. the probability of finding the object in a given region was directly proportional to the size of the region. We believe that is a good general a priori, given that to consider other types of pdfs, the type of object should be taken into account. However, in a scenario where one or more objects are searched several times, it should be possible to start the search assuming an uniform pdf and modify it according to the places where the objects are found. We leave this problem for future work.

Acknowledgments

This research was partially funded by CONACyT project 56754-F1, NSF-CONACyT Project J110.534 and CONCYTEG project 07-02-K662-097.

References

1. Sarmiento, A., Murrieta-Cid, R., Hutchinson, S.: A Sample-based Convex Cover for Rapidly Finding an Object in a 3-D environment. In: Proc. IEEE Int. Conf. on Robotics and Automation (2005)
2. Sarmiento, A., Murrieta-Cid, R., Hutchinson, S.: An Efficient Strategy for Rapidly Finding an Object in a Polygonal World. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (2003)
3. Ishiguro, H., Sogo, T., Barth, M.: Baseline Detection and Localization for Invisible Omnidirectional Cameras. International Journal of Computer Vision 58(3), 209–226 (2004)
4. Hert, S., Tiwari, S., Lumelsky, V.: A terrain-covering algorithm for an auv. Autonomous Robots 3, 91–119 (1996)
5. Chin, W.P., Ntafos, S.: Optimum watchman routes. Information Processing Letters 28, 39–44 (1988)
6. Goodman, J.E., O'Rourke, J. (eds.): Handbook of Discrete and Computational Geometry. CRC Press, Boca Raton (1997)
7. Hsu, D., Latombe, J.C., Motwani, R.: Path planning in expansive configuration spaces. In: Proc. IEEE Int. Conf. on Robotics and Automation (1997)
8. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation 12(4) (June 1996)
9. Latombe, J.-C.: Robot Motion Planning. Kluwer Academic Publishers, Dordrecht (1991)
10. Leven, P., Hutchinson, S.: A Framework for Real-time Path Planning in Changing Environments. Int. Journal Robotic Res. 21(2), 999–1030 (2003)
11. O'Rourke, J.: Art Gallery Theorems and Algorithms. Oxford University Press, Oxford (1987)
12. Simeon, T., Laumond, J.P., Nissoux, C.: Visibility based probabilistic roadmaps. Advanced Robotics Journal 14(6) (2000)

Extraction and Specialization of Geo-spatial Objects in Geo-images Using Semantic Compression Algorithm

G. Guzmán, S. Levachkine, M. Torres, R. Quintero, and M. Moreno

Centre for Computer Research, National Polytechnical Institute, Mexico City, Mexico
{jguzmanl,sergei,mtorres,quintero,marcomoreno}@cic.ipn.mx
http://geo.cic.ipn.mx, http://www.cic.ipn.mx

Abstract. This paper describes an object oriented methodology for the semantic extraction of a geo-image, which is defined by a set of natural language labels. The approach is composed of two main stages: *analysis* and *synthesis*. The analysis stage detects the main geographic components of a geo-image by means of the color quantification, geometry and topology of the geospatial objects. The result of this stage is a set of geo-images with intensities that are approximately uniform. The synthesis stage extracts the main geographic objects that have been identified and a labeling process is made in two levels (general and specialized). The aim of the labeling process is to associate a label of the thematic to each region, taking into account the RGB characteristics of the geo-image. In order to specialize each geographic object, we have proposed a specialization algorithm that considers geometric and topologic relations among them, represented in geographic application domain ontology. As a result, the set of labels describes the *semantics* of a geo-image.

1 Introduction

Nowadays geospatial data of remote images (geo-images) are very useful, because with them it is possible to obtain information for planning, construction or simulation of natural disasters tasks. Some of the challenges related to geo-images that researchers are currently addressing are the following: automatic analysis, recognition, classification, object decomposition, among others.

Basically the main goal of any image processing technique is to make a partition, label assigning or class-id, with the purpose to describe the main regions and determine the geographic objects represented by the geo-image.

For this process, it is possible to apply approaches of Remote Sensing (RS) and Digital Image Processing (DIP). To date, algorithms can be classified in two major categories: pixel based and object oriented methodologies. The most of DIP or RS approaches, such as hierarchical analysis [10], segmentation [6, 7, 9, 13], wavelets, or others techniques [2, 3, 5], are centered at the first category. The most important limitation of these algorithms is the following: its result is an image or a set of sub-images, requiring pre-processing and post-processing stages, and uses a single or reduced set of variables to determine the partition. However, we, the humans, are best understood the assignation of labels or concepts (semantic approach) not the use of regions or clusters (numerical approach).

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 573–584, 2008.

On the other hand, the object oriented approaches try to use a priori information that is not presented in the geo-image [14]. In this case the main limitation is the use of traditional DIP algorithms to extract the geographic objects.

To determine the semantics of a geo-image, it is necessary more information that is not explicitly located in the image. For instance, the concept *island* is defined as "portion on land surrounded by a water body". Another case, the concept *lake* is defined as follows "it is a water body insides a land". This is the knowledge (the objects and relations between them) that human beings use to determine the contents of the visual information [15].

In conclusion, we have detected two main problems with current methods: the limitations of use one or a reduced set of properties to establish the main objects in a geo-image, and the form to specialize them. In this paper, we propose an object oriented algorithm that consists of two main stages: *analysis* and *synthesis* to describe the *semantics* of the geo-images, i.e. the image objects and their labels.

The rest of the paper is organized as follows: in Section 2 we describe some aspects of the transformation from RGB into isotropic space; that is the semantic compression algorithm; Section 3 describes the proposed methodology that is composed of the analysis and synthesis stages. Section 4 depicts the results obtained for some geo-images. Conclusions and future work are pointed out in Section 5.

2 Transformation from RGB into Isotropic Space

The semantic compression algorithm quantifies different characteristics in an isotropic space of a segment set which is represented by means of a dynamic tree (hierarchical segments) [1]. The initial assertion is that the number of segments is equal to the number of discretional elements (pixels) of the geo-image. This condition is necessary because there is not a priori knowledge about the structure or form of the geographic objects contained in the geo-image. When two segments satisfy the adjacent condition and some property or characteristic measured between them is similar, then these segments will be merged. A new node is created and associated with the involved segments in the dynamic tree to represent the fusion of the segment.

The hierarchical segments are generated by step-to-step integration of similar areas, that is, in a recursive form. The selection of the criteria that allows for making a segment fusion is limited by a set of characteristics that are computed for each segment in the fusion process.

The segment characteristics can be classified in two groups: attributes and properties. The attributes are a primary set of segment characteristics, dynamically estimated and stored for all segments at any level of their representation in the dynamic tree. The properties are numerical segment characteristics, determined as an output of data conversion and selected in function of the processing stage and the problem context.

The set of characteristics for our context is sorted according to the complexity order: global characteristics (for all geo-image), local characteristics (inside the neighborhood for one segment), integral intensity (sum of all intensities), number of pixels, first and second order moments (computed with respect to the center of the segment), no additive perimeter and description of adjacent segments.

This list represents the geometric and intensity properties, such as: pixel intensities range, average intensity and invariant moments. To determine the numerical expressions to quantify the segment properties for the construction of the object hierarchy, it is necessary to work in a different space applying a space transformation that allows us to convert the objects in isotropic objects [11, 12].

If we suppose that it is possible to determine the orientation of some image object, then it is necessary to define a new adequate coordinate system to it. By using this new coordinate system, we can make an equalizing of the axis scale. In this coordinate system, the areal objects are described by an invariant variable equal to the media square root of the size.

If a set of punctual objects does not intersect with a line, these objects will form a non-degenerative object, which is performed by a linear transformation. It is converted into an isotropic object that has a uniform value for the media quadratic square of the size. An isotropic object composed of n −points is defined by one rule: the media square of the distance of the isotropic object points to the gravity center of any line which does not depend on line pending.

In a linear Euclidean space (u, v) induced by means of linear combinations (of rows) from the coordinates systems (x, y), the second order moments computed with respect to the center of inertia are obtained as scalar products (see Equation 1).

$$I_x \equiv (u,u), I_y \equiv (v,v), I_{xy} \equiv (u,v), \tag{1}$$

where the isotropic figures are represented by a pair of orthogonal vectors of the same length, this condition is expressed in terms of the second order moments that is equivalent to:

$$\left.\begin{matrix}(u,v)=0\\(u,u)=(v,v)\end{matrix}\right\} \Leftrightarrow \begin{cases} I_{xy}=0 \\ I_x = I_y \end{cases} \tag{2}$$

Introducing the independent parameters Δ and γ, they are related with the second order moments by means of the Equation 3:

$$\Delta,\gamma = \begin{cases} \cos\Delta = \dfrac{I_{xy}}{\sqrt{I_x I_y}} \\ \sinh\gamma = -\dfrac{1}{2}\left(\sqrt{\dfrac{I_x}{I_y}} - \sqrt{\dfrac{I_y}{I_x}}\right) \end{cases} \tag{3}$$

The expression of hyperbolic sinus can be extended using the mathematical properties of roots, obtaining the Equation 4.

$$\sinh\gamma = \frac{1}{2}\left(\frac{\sqrt{I_y}}{\sqrt{I_x}} - \frac{\sqrt{I_x}}{\sqrt{I_y}}\right) = \frac{1}{2}\left(\frac{\sqrt{I_y}\sqrt{I_y} - \sqrt{I_x}\sqrt{I_x}}{\sqrt{I_x}\sqrt{I_y}}\right) = \frac{1}{2}\left(\frac{I_y - I_x}{\sqrt{I_x I_y}}\right) \tag{4}$$

By using the properties $(\sin\theta)^2 + (\cos\theta)^2 = 1$ and $(\cosh\theta)^2 - (\sinh\theta)^2 = 1$, for variable, we have that:

$$\sin\Delta = \sqrt{1-(\cos\Delta)^2} = \sqrt{1 - \frac{(I_{xy})^2}{I_x I_y}} \tag{5}$$

Due to the properties of the hyperbolic trigonometric functions, we have the Equation 6:

$$\cosh\gamma = \sqrt{1+(\sinh\gamma)^2} = \sqrt{1 + \frac{1}{4}\frac{(I_y - I_x)^2}{I_x I_y}} \tag{6}$$

The non-degraded objects are associated to a value of γ, when $\sinh\gamma \neq 0$. On the other hand, a value is near zero in a variable denoted by d, which represents the isotropic objects. This value is computed by means of the Equation 7:

$$d = \sqrt{(\cos\Delta)^2 + (\sinh\gamma)^2} \tag{7}$$

Any non-degraded isotropic object will be transformed in isotropic one by means of a non-linear transformation, denoted by W. This transformation in the space $u \times v$ is reduced to an orthogonalization and equalization of the length of u and v vectors that in the initial plane $x \times y$ means a stretching of the main axis and the compression of other. The W transformation is denoted by an equalization of the I_x and I_y moments and the assignation of a value equal to zero to the union of moments (see Equation 8).

$$\left.\begin{aligned} W\{u\} &= e^{\theta}(u\cos\varphi - v\sin\varphi) \\ W\{v\} &= e^{-\theta}(-u\sin\varphi - v\cos\varphi) \end{aligned}\right\} \Rightarrow \begin{cases} \tilde{I}_{xy} = 0 \\ \tilde{I}_x = \tilde{I}_y \end{cases} \tag{8}$$

The φ angle that determines the transformation of the u and v vectors is associated with the rotation of initial plane and defined by Equation 9.

$$\varphi = \begin{cases} \sin 2\varphi = \sigma\dfrac{\cos\Delta}{d} \\ \cos 2\varphi = -\sigma\dfrac{\sinh\gamma}{d} \end{cases}, \quad \sigma = \pm 1 \tag{9}$$

The formula of the φ angle determines the axis orientation of the non-isotropic objects with respect to the initial configuration. In particular, the symmetric figures allow us to find the direction of symmetry axis. The hyperbolic parameter θ is defined by the Equation 10:

$$\theta = \tanh 2\theta = -\sigma\frac{d}{\coth\gamma} \tag{10}$$

This parameter (θ) describes in logarithmic scale the radio of the linear dimensions:

$$\theta = \frac{1}{2}\sigma\ln(width \times large) \tag{11}$$

The object dimensions are indistinctly calculated from the orientation, using the media quadratic of the point distances to the axis:

$$large^2 = \frac{\sqrt{I_x I_y}}{n}(\cosh\gamma + d)\,,\ width^2 = \frac{\sqrt{I_x I_y}}{n}(\cosh\gamma - d) \tag{12}$$

The square of the invariant linear size of an isotropic object determines its area that corresponds with the area of the initial object (non-isotropic) and it is equal to the product of $large$ and $width$ variables:

$$size^2 = area = \frac{\sqrt{I_x I_y}}{n}|\sinh\Delta|, \qquad size = \sqrt{\frac{\sqrt{I_x I_y}}{n}|\sinh\Delta|} \tag{13}$$

The number of points, the $area,\ large,\ width$ variables, the object invariant $size$, the trigonometric and hyperbolic parameters that allow for transforming some objects in isotropic ones, are estimated for the whole geo-image or in the fusion process. To establish that W transformation from the bi-dimensional space $u \times v$ matches with the transformation of the initial plane $x \times y$, it is sufficient to show it as the product of the orthogonal

transformation V with the L set[1], which allows us to consider these planes as a vector set. The points of the figure are described by w_i vectors:

$$w_i = x_i e_x - y_i e_y, \tag{14}$$

where e_x and e_y are orthonormal vectors defined by:

$$e_x, e_y = \begin{cases} e_x = \dfrac{W\{v\}}{\sqrt{\sqrt{I_x I_y}|\sin\Delta|}} \\ e_y = \dfrac{W\{u\}}{\sqrt{\sqrt{I_x I_y}|\sin\Delta|}} \end{cases} \tag{15}$$

The u y v vectors in the e_x, e_y coordinate can be expressed by the Equation 16:

$$\begin{aligned} u &= -\sqrt{n} \cdot size\left(e_x \cdot e^{\theta} \cdot \sin\varphi + e_y \cdot e^{-\theta} \cdot \cos\varphi\right) \\ v &= -\sqrt{n} \cdot size\left(e_x \cdot e^{\theta} \cdot \cos\varphi - e_y \cdot e^{-\theta} \cdot \sin\varphi\right) \end{aligned} \tag{16}$$

The rotation (V) and the Lorentz transformation (L) are defined by:

$$\begin{aligned} V &= \begin{cases} V(e_x) = e_x \cos\varphi - e_y \sin\varphi \\ V(e_y) = -e_x \sin\varphi + e_y \cos\varphi \end{cases}, \\ L &= L\{w\} = e^{-\theta}(w, e_x)e_x - e^{\theta}(w, e_y)e_y \end{aligned} \tag{17}$$

where w is a random vector of the $x \times y$ or $u \times v$ planes. Thus in the case of a passive interpretation of the coordinate's transformation W, we have that:

$$w_i = x_i e_x - y_i e_y \equiv \left(w_i, W^{+^{-1}}\{e_x\}\right)W(e_x) + \left(w_i, W^{+^{-1}}\{e_y\}\right)W(e_y) \tag{18}$$

V and L are the components of the decomposition to obtain the Equation 19:

$$W = VL = \begin{cases} V^{+} = V^{-1} \\ L^{+} = L \end{cases}, \tag{19}$$

where the plus sign denotes the conjugate and the -1 exponent refers to the inverse. The new coordinates $(w_i, VL^{-1}\{e_x\})$, $\left(w_i, VL^{-1}\{e_y\}\right)$ are the result of the plane deformation and describes the $W\{u\}, W\{v\}$ multidimensional components, while, for an active interpretation of the W coordinates transformation the previous expression compliant with the inverse transformation.

Furthermore, any non-isotropic object composed of *n*-points in the $x \times y$ plane is compared in the same plane with the pair of u, v vectors which are represented as columns in the same coordinate space $(u \to x, v \to y)$. The u, v vectors are determined up to a $\pi/2$ precision. These vectors are transformed in orthogonal ones with the same length applying the transformation that converts the objects into isotropic.

By using an algorithm that generates the hierarchical compact structure considering the geometric attributes, it is possible to obtain the representation of adaptative objects for a real image with texture, patterns, and so on. All figures with a high grade of similarity will be merged when their respective isotropic representations are compared, making a logical union of the adjacent segments. To select the pair of the segments to merge, an estimation of the grade of difference between the composite images and isotropic ones is performed. To do this, the parameter d defined in Equation 7 can be used. The obtained result is a compact hierarchy of all image segments.

[1] L is a symmetric conjugate of V.

It is necessary to consider the case when a fusion of two objects composed of n_1 and n_2 points is done, the second order moments I_x, I_y, I_{xy} satisfy the next properties:

$$I_x = I_x^{(1)} + I_x^{(2)} + \frac{n_1 n_2}{n_1+n_2}\delta x^2, I_y = I_y^{(1)} + I_y^{(2)} + \frac{n_1 n_2}{n_1+n_2}\delta y^2$$
$$I_{xy} = I_{xy}^{(1)} + I_{xy}^{(2)} + \frac{n_1 n_2}{n_1+n_2}\delta x \delta y\,, \tag{20}$$

where the 1 and 2 subscripts denote the number of segments, $\delta x\,, \delta y$. In addition they describe the relative offset to the inertia center, that is, the distance between the centers of each segment.

When the compact hierarchy is build, the value of d is expressed by the y φ parameters that are obtained using the new moments (Equation 20). If the conditions $I_{xy} = 0$ and $I_x = I_y$ are accomplished, the new object is described by the relation $d = 0$ and apparent isotropic object. The result of superposition of two isotropic objects is a new isotropic object *iff* the centers of inertia coincide.

Finally, the set of characteristics and parameters is composed of: intensity (int)[2], average intensity (ABS), distance (d), large (L), width (W), size (Sz), area (S), $sin\,\Delta$ (Ss), $cos\,2\gamma$ $(C2f)$ and $sin\,2\gamma$ $(S2f)$.

3 Proposed Algorithm

The semantic decomposition algorithm is composed of two stages: analysis and synthesis. In this section, we describe these stages.

Analysis Stage

It consists of applying the semantic compression algorithm; the input is the geo-image source and the result is a set of geo-objects[3], each one is described by uniform intensities. The semantic compression algorithm consists of several steps, which generate the dynamic tree. A recursive mode is used to merge all the adjacent segments, according to similarity of some attribute or property. A normalization process is made and it finalizes generating the set of geo-objects with uniform intensity. The merging segment is based on a similarity condition (according to the semantic compression threshold[4]). It should be independently processed for each characteristic. By using the complete set of characteristics and parameters at the end of the semantic compression algorithm, the result will be a total of 10 geo-images. The steps involve in this algorithm are the follows:

1. *Creation and initialization of dynamic tree.* Create a dynamic tree (B) for each characteristic or parameter to be quantified. This structure has a root node and a total of $M \times N$ *child nodes*, in which M and N denote respectively the number of rows and columns of the source geo-image. For each *child node*, we assign an index based on pixel position, according to the absolute position of the element represented in the geo-image (see Equation 21).

$$index = \big((x+1)\times M\big) + \big((y+1)\times N\big) \tag{21}$$

[2] The intensity is not abbreviated with the symbol (I) to avoid confusion with the geo-image symbol (I).

[3] A geo-object is a subset of geo-image, which has some meaning for the user. For instance "the sea" in the satellite image of the Earth.

[4] The semantic compression threshold is referred also as similarity threshold.

2. *Merging of homogeneous segments*. It consists of merging all adjacent segments with the same value of intensity, creating a new father node and assigning as child the merged segments:

$$s_l = s_i \cup s_j = \{p_1, p_2, \dots, p_m\} \cup \{q_1, q_2, \dots, q_n\} \tag{22}$$

3. *Repeat condition*. From this step, we apply a recursive process until the number of fusion will be equal to zero (invariant geo-image).

4. *Characteristic or parameter quantification*. Let (B) be a dynamic tree that represents the segments of the geo-image according to some characteristic or parameter (c_s). It is necessary to compute c_s for each segment that is not associated with other segment (that is, they do not have a common father), and for each segment that describes a merging (father node).

5. *Tolerance computation*. First, we need to determine the maximal and minimal value of numerical c_s, obtained in the previous step:

$$c_{s\_max} = max\left(c_{s_1}, c_{s_2}, \dots, c_{s_n}\right), c_{s\_min} = min\left(c_{s_1}, c_{s_2}, \dots, c_{s_n}\right) \tag{23}$$

With this pair of values, we can find the max difference between the c_s:

$$\Delta c_{max} = c_{s\_max} - c_{s\_min} \tag{24}$$

To define the tolerance, we determine the product of the semantic compression threshold[5] ($k_{cs} \in [0 \dots 100]$) and Δc_{max} :

$$k_c = \Delta c_{max} \times k_{cs} \tag{25}$$

6. *Merging*. Join all adjacent segments that satisfy the Equation 26:

$$s = s_i \cup s_j, \ \ \forall\, i \neq j \mid \left(\left|c_{s_i} - c_{s_j}\right| < k_c\right) \tag{26}$$

7. *Normalization*. The final output is a geo-image in which the intensity for every pixel is normalized in the range $[0, \dots, 255^3]$, according to the value of c_s for the segment that the pixel belongs.

This algorithm determines the merging, according to specific characteristic or parameter. In consequence, the association between segments may be different with respect to other characteristic or parameter. In fact, the user needs to choose the semantic compressed geo-image. Generally, after the execution of the algorithm, the desired simplification is not reached, due to the complexity of geographic objects. Thus, it is necessary to apply again the algorithm, using as new input, the manually selected semantic compressed geo-image. The goal is to execute the semantic compression algorithm, until each significant geographic object is described by a uniform RGB value. Formally, the semantic compression stage consists of determining a semantic compression string:

$$C_{cs}(I_o, I_{ccs}) = \left\{\left(I_o, k_{cs_1}\right), \left(I_1, ccs_1, k_{cs_2}\right), \left(I_2, ccs_2, k_{cs_3}\right), \dots \left(I_{i_{cs}-1}, ccs_{i_{cs}-1}, k_{cs_{i_{cs}}}\right)\right\}, \tag{27}$$

where:

$\left(I_{i-1}, ccs_{i-1}, k_{cs_i}\right)$ are the input parameters used in the *i*-iteration number.
k_{cs_i} denotes the similarity threshold used in the *i*-iteration number.
ccs_{i-1} is the characteristic or parameter employed to simplify the I_{i-1} geo-image.
I_0 is the original geo-image.
I_{ccs} is the final geo-image selected at the end of the i_{cs}-iteration.

The final geo-image obtained in this stage will be referred as the I_A geo-image.

[5] With this equation, it is possible to use the same semantic compression threshold with all characteristics or parameters, because the tolerance adjusts the variations determined in each case.

Synthesis Stage

This stage has the purpose of obtaining the semantics of the selected geo-objects from the analysis stage. It is composed of the algorithms: region extraction, recognition of geographic objects and specialization of geographic objects.

Region extraction. The output geo-image obtained with the semantic compression algorithm is described by a set of uniform intensities, which can be used for the recognition process. In this algorithm, we retrieve the original intensities of geographic objects to process them and assign to each one a thematic label in the next phase. Basically, we need to use a mapping function to extract each region with original RGB values, then the result of this process is a list of regions, where the number of regions is equal to the different intensities obtained in the semantic compression algorithm. Let i be a class-id where $i = \{1,2, \dots, n_{rcs}\}$. The algorithm extracts all homogeneous regions, which have been assigned to the i-class. By using these regions, we generate a new geo-image, denoted by I_{T_i}, in terms of the Equation 28:

$$I_{T_i} = \{(x,y) \in I_A \mid N_c(x,y) = i\}, \tag{28}$$

where $N_c(x,y)$ is a function that determines the number of class that the pixel $p(x,y)$ and belongs in the I_A geo-image; in consequence $Range(N_c) = \{1,2, \dots n_{rcs}\}$. The second operation is used to replace the intensities of region to retrieve original values, using the Equation 29:

$$I_{T_i}(x,y) = \langle R\big(I_O(x,y)\big), G\big(I_O(x,y)\big), B\big(I_O(x,y)\big) \rangle \forall\, p(x,y) \in I_{T_i} \tag{29}$$

Thus, the output of the algorithm gives a total of n_{rcs} geo-images $I_T = \left\{I_{T_1}, I_{T_2}, \dots, I_{T_{n_{rcs}}}\right\}$, in which each geo-image describes all geographic objects (regions) of the same thematic.

Recognition of geographic objects. Note that there is not a unique set of thematic layers for the recognition of geographic objects; this set always depends on the application and personal requirements. Formally, a thematic layer (water body, land, among others) is a label that describes similar geographic objects in the RGB space. The set of thematic layers is composed of all labels in which the geographic objects need to be classified:

$$T = \left\{t_1, t_2, t_3, \dots, t_{n_T}\right\}, \tag{30}$$

where n_T is the number of thematic layers defined in the synthesis stage. By using this set, we do a coarse scale labeling of geographic objects. In tests, we define prototype vectors by means of training, using a bank of 20 regions for each thematic layer. Later, by applying a recognition algorithm (for instance, minimal Euclidean distance), we determine the most similar thematic layer for each I_{T_i}. The result of this algorithm is a label set defined as $T_{I_o} = \left\{eg_1, eg_2, \dots, eg_{n_{rcs}}\right\}$.

Specialization of geographic objects. This algorithm has the goal of specializing each label obtained in the recognition process. For instance, let a region be with the *water body* label, we interest to determine that this water body represents a *river*, *lake*, or other concept. In other case, a *land* region is a *continent* or an *island*. Each region $\left\{R_1, R_2, \dots, R_j\right\}$ of some I_{T_i} geo-image describes a geographic object $\left\{O_1, O_2, \dots, O_j\right\}$ to specialize them. We propose to use an application domain ontology [4, 8], in which the relations between the objects need to be determined by means of a set of Digital Image Processing operators. For instance, the lake concept is described as "portion of land surrounded by a water body". The surrounded relation can be determined using the intersection operator. One

important concept in the ontology is the concept "other"; if the size of some region is less or equal to specific threshold, we classify this object as "other". This constraint is necessary because the number of pixels that describes this region is not enough to make a recognition or specialization. Finally, the semantics description of geo-image I_O is a set of specific labels obtained at the end of the specialization algorithm:

$$S_{I_o} = \{sl_1, sl_2, \ldots, sl_{n_o}\}, \tag{31}$$

where n_o is the number of objects identified in I_O and sl_i is a specific label.

4 Tests and Results

In this section we depict the obtained results, using the proposed algorithm. The test images are illustrated in Figure 1.

Fig. 1. Geo-images used in tests; (a) Case 1: Island I, (b) Case 2: Island II, (c) Great-Lakes, (d) Mexico

The semantic compression strings that produced best results for each geo-image are enlisted in Table 1. An important characteristic of this approach is that we can determine different semantic compression strings for one geo-image, and we obtained the same result. The geo-images obtained at the end of each semantic compression stage are sketched out in Figure 2. We note the homogenization process represented by means of uniform intensities conform the number of iteration increments.

We use the media, median and standard deviation in each color component and the Euclidean distance to classify the regions and assign the label of thematic layer. The thematic layer used in this test was $T = \{body\ water, land\}$. There are two great categories of land, the first that describes deserted or mountain regions and the second describes vegetation zones, we have two different vector prototypes for the second thematic layer. The geo-images generated after applying the extraction algorithm are shown in Figure 3.

Table 1. Semantic compression strings used in each geo-image

Geo-image	Semantic Compression String
1	$C_{cs}(I_o, I_{Ss}) = \{(I_o, 50), (I_1, ABS, 50), (I_2, S2f, 50), (I_3, L, 50), (I_4, Ss, 50)\}$, or $C_{cs}(I_o, I_{Ss}) = \{(I_o, 50), (I_1, ABS, 50), (I_2, S, 50)\}$
2	$C_{cs}(I_o, I_{Ss}) = \{(I_o, 21.5), (I_1, int, 22.5), (I_2, S, 22.5), (I_3, S, 22.5)\}$
3	$C_{cs}(I_o, I_{int}) = \{(I_o, 20), (I_1, int, 20), (I_2, int, 20), (I_3, int, 20), (I_4, int, 05)\}$
4	$C_{cs}(I_o, I_L) = \{(I_o, 21.5), (I_2, int, 2.5), (I_3, int, 1.5), (I_4, D, 90.0)\}$

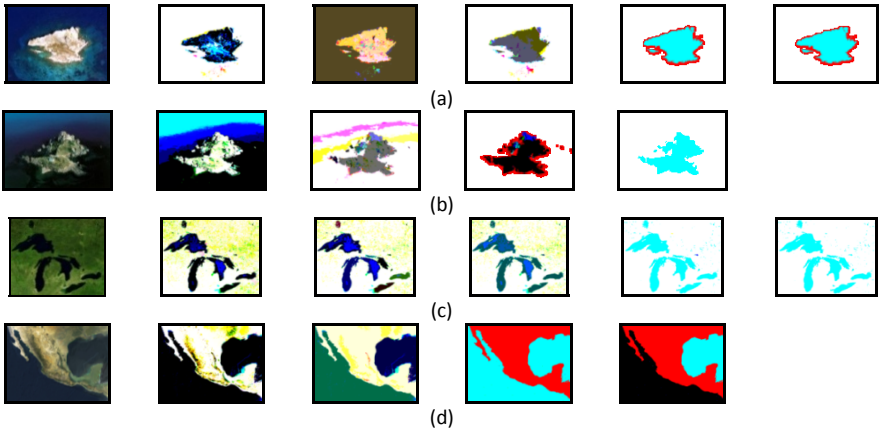


Fig. 2. Geo-images obtained applying the semantic compression strings from Table 1 for original geo-images; (a) Geo-image 1, (b) Geo-image 2, (c) Geo-image 3, (d) Geo-image 4

In Table 2, we present the set of labels obtained in the recognition and specialization algorithms. Note that only one instance of the same concept is described. For instance, in the case of the Great Lakes geo-image, we have several lakes, but only one label appears.

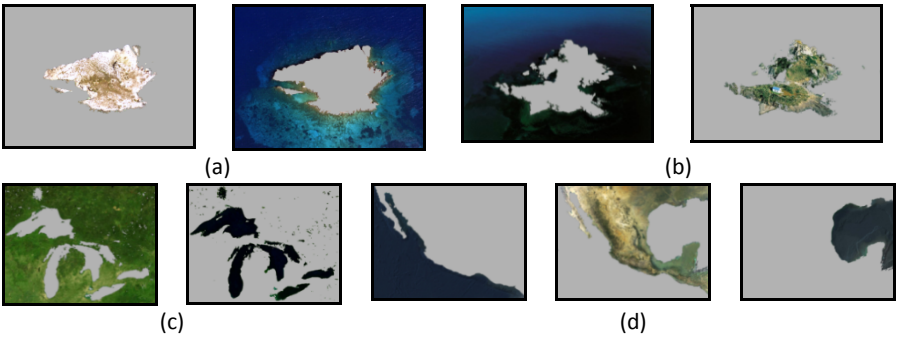


Fig. 3. Result of the algorithm for region extraction applied to the geo-image obtained in the analysis stage (a) Island I, (b) Island II, (c) Great Lakes, (d) Mexico

Table 2. Results of recognition and specialization algorithms

Geo-image	Label results
1	$T_{I_O} = \{land, body\ water\}$ $S_{I_O} = \{island, ocean\ or\ sea, others\}$
2	$T_{I_O} = \{body\ water, land\ (green\ area)\}$ $S_{I_O} = \{ocean\ or\ sea, island, others\}$
3	$T_{I_O} = \{land\ (green\ area), body\ water\}$ $S_{I_O} = \{land, INLAND, islet\}$
4	$T_{I_O} = \{land, body\ water\}$ $S_{I_O} = \{land, sea\}$

5 Conclusions

In this paper, a general decomposition methodology to obtain the meaningful geographic objects in geo-images is described. In the analysis stage, we show a new RGB-isotropic space transformation that serves to quantify some attributes and geometric properties of the segments. The main goal of this stage is to generate *semantic compression strings* to simplify the intensities presented in the geo-image, until each relevant geographic object is described with a homogeneous RGB value. In the synthesis stage, we obtain the *semantics* of the geo-image using two labeling approaches: 1) applying a Euclidean classifier and 2) using application domain ontology. We consider that relations between objects are very important to correctly specialize the geographic objects. In consequence, we can obtain a good specialization similar that human beings perceive by means of their cognitive system. Future work is related to the study of some particular geographic domains in order to determine semi or automatic compression strings. Additionally, we are centered this research at comparison of the performance of our methodology with some commercial systems, but it is important to note that there is not similar system, which considers all the *semantic* aspects proposed in this work.

Acknowledgements

The authors of this paper wish to thank CIC, SIP, IPN, and CONACYT for their support.

References

1. Adams, N.J., Williams, C.K.I.: Dynamic trees for image modeling. Image Vision Computing 21(10), 865–877 (2003)
2. Angulo, J., Serra, J.: Mahematical Morphology in Color Spaces Applied to the Analysis of Cartographic Images. In: Proceedings of the Second International Workshop on Semantic Processing of Spatial Data (2003)
3. Bezdeck, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
4. Borst, Construction of Engineering Ontologies, Ph.D. Thesis, University of Tweenty, Enschede, NL – Centre for Telematica and Information Technology (1997)
5. Bradshaw, B.: Semantic Based Image Retrieval: A Probabilistic Approach. ACM Multimedia, 167 – 176 (2000)
6. Byung-Gyu, K., Jae-Ick, S., Dong-Jo, P.: Fast image segmentation based on multiresolution analysis and wavelets. Pattern Recognition Letters 24, 2995–3006 (2003)
7. Chen, J., Pappas, T.N., Mojsilovic, A., Rogowitz, B.E.: Adaptive Perceptual Color-Texture Image Segmentation. IEEE Transactions on Image Processing, 1524–1536 (2005)
8. Corcho, O., Fernández-López, M., Gómez-Pérez, A.: Methodologies, tools and languages for building ontologies. Where is the meeting point? Data & Knowledge Engineering 46, 41–64 (2002)
9. Din-Yuen, C., Chih-Hsueh, L., Wen, H.: Image Segmentation with Fast Wavelet-Based Color Segmentation and Directional Region Growing. IEEE Transactions Informatics and Systems E88-D(10), 2249–2259 (2005)

10. Huang, J., Ravi Kumar, S., Zabih, R.: An Automatic Hierarchical Image Classification Scheme. ACM Multimedia, 219–228 (1998)
11. Levachkine, S.: Raster to Vector Conversion of Color Cartographic Maps. In: Proceedings of 5th International Workshop Graphics Recognition, Recent Advances and Perspectives, GREC 2003, pp. 50–62 (2003)
12. Levachkine, S., Velázquez, A., Alexandrov, V., Kharinov, M.: Semantic Analysis and Recognition of Raster-Scanned Color Cartographic Images. In: Proceedings of 3th International Workshop Graphics Recognition, Recent Advances and Perspectives, GREC 2001, pp. 178–189 (2001)
13. Liu, J., Yang, Y.H.: Multiresolution Color Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 16, 689–700 (1994)
14. Mueller, M., Segl, K., Kaufmann, H.: Edge- and region-based segmentation technique for the extraction of large, man-made objects in high-resolution satellite imagery. Pattern Recognition 37(8), 1619–1628 (2004)
15. Fonseca, F., Egenhofer, M., Davis, C., Câmara, G.: Semantic Granularity in Ontology-Driven Geographic Information Systems. In: AMAI Annals of Mathematics and Artificial Intelligence – Special Issue on Spatial and Temporal Granularity, vol. 36(1, 2), pp. 121–151 (2002)

A Hybrid Rough K-Means Algorithm and Particle Swarm Optimization for Image Classification

Chih-Cheng Hung and Hendri Purnawan

Southern Polytechnic State University
Marietta, GA 30060-2896 USA
{chung,hpurnawa}@spsu.edu

Abstract. This paper proposes a hybrid rough K-means algorithm for image classification. The rough set theory is used to establish the lower and upper bound for data clustering in the K-means algorithm. Then, the particle swarm optimization (PSO) is employed to optimize the solutions of the rough K-means algorithm. The combined algorithm is called the Rough K-means PSO algorithm. Experimental results show that the proposed algorithm performs better and improves the classification in the blurred and vague areas of test images.

Keywords: K-means Algorithm, Rough K-means, Particle Swarm Optimization, Image Segmentation.

1 Introduction

In the real world, data representation is most often imperfect, in the sense that the data may be either incomplete or redundant. The word "uncertainty" will be used in this paper. Philosophers, logicians and mathematicians have dealt with this problem for a long time. The problem of imperfect knowledge is also becoming an important topic for computer scientists, who are engaged in the artificial intelligence research, especially in the area of knowledge discovery from database, expert systems, and pattern recognition.

Pawlak introduced rough set theory, which deals with the uncertainty of data, in the early 1980's [1]. The theory is a new approach to vagueness and uncertainty. Although rough set theory overlaps with other theories, such as statistic, evidence theory and fuzzy sets, it has been considered a self-sufficient discipline. The fuzzy sets theory has actually been increasingly used collaboratively with rough set theory in classification and clustering applications. Fuzzy set is another approach to deal with vagueness. One example of combining the rough and fuzzy concepts for image analysis was proposed in [2]. The paper shows that the result was encouraging.

Rough set theory has been used in many applications, among which are knowledge discovery in database, predictive maintenance, medicals, engineering and etc [3]. The rough set theory has one main advantage for data analysis that is no preliminary data, such as "the value of possibility" used in fuzzy set theory is needed.

The K-means algorithm is a simple and efficient clustering algorithm. It works by minimizing the squared error function. In spite of its simplicity, the K-means

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 585–593, 2008.

algorithm is plagued with its inherent problem; that it can be trapped into local optimal error. In other words, instead of finding the global optimal solution, it frequently finds local optimal solution. To overcome this drawback, the rough set theory and particle swarm optimization (PSO) are proposed to improve the performance of the K-means algorithm.

The paper is organized as follows. Section 2 gives a brief introduction to rough set theory. Section 3 reviews the K-means clustering algorithm. Section 4 introduces particle swarm optimization. Section 5 gives the detail of the Rough K-means PSO algorithm for image segmentation. Section 6 shows the experimental results. The conclusion then follows.

2 Rough Set Theory

Rough set theory is a mathematical tool to dealing with vagueness and uncertainty of data. The theory consists of finite sets, equivalence relation and cardinality concepts [1]. However, as the theory matures and more applications reap the benefits of the concept, an abundant of related theorems and algorithms are incorporated to extend the rough set theory.

Rough set is mainly used in data analysis in the table format. The approach in general would be to pre-process the data in the table and then discern them. Reducts are extracted by using algorithms. Rules are then generated based on the reducts. Rough set does not support analog data, which are expressed as attributes in the table; hence discretization must be performed on analog attribute values in order to process the data in the table.

3 The Rough K-Means Algorithm

The K-means algorithm (hereafter referred to as K-means) is one of the simplest and efficient unsupervised learning algorithms for clustering problems such as image segmentation [4, 9]. In this algorithm, random centroids are assigned and repeatedly modified through the iterations in order to minimize the squared error function. Suppose there are N pixels in an image to be classified into m centroids. Each pixel v_i, where $1 \le i \le N$, is assigned to one of the centroids c_j, where $1 \le j \le m$, based on squared Euclidean distance of each pixel to each centroids.

$$d(v,c) = \sum_{i=1}^{N} \sum_{j=1}^{m} (v_i - c_j)^2 \tag{1}$$

Upon the completion of the assignment, each of new centroids is calculated using

$$c_j = \frac{\sum_{v \in c} v_i}{n} \tag{2}$$

where $1 \le i \le N$, and n is the number of pixel in cluster c_j. The process ends when c_j stabilizes. The weakness of K-means is that it depends on initial selection of

centroids and it may be trapped into local optimal results. However, running the algorithm repeatedly and randomly selecting different sets of clusters centroid may offset the problem. It is shown that the local optimal results may also be avoided by assigning the centroids based on distribution of the pattern in the histogram of an image [5].

4 Particle Swarm Optimization

The particle swarm optimization (PSO) was originally introduced by Kennedy and Eberhart [6]. The algorithm was inspired by a sociological behavior of a flock of birds searching for food. Each member of the flock moves at the direction and speed that are affected by its own previous state and that of the entire flock.

In essence, the PSO algorithm consists of a swarm (collection) of particles searching through the solution space. Each particle holds information that can potentially becomes the solution. Each particle has position and velocity that are mutually affecting those of other particles'. Each particle will adjust its parameters closer to the swarms' best outcome, yet weighting in its own best experience. Therefore, at any instance, each particle maintains x_i (the current position of the particle), v_i (the current velocity of the particle), y_i (the personal best position of the particle (*pbest*) or the best position visited so far by the particle) and $\hat{y}$ (the global best position of the swarm (*gbest*) or the best position visited so far by the entire swarm).

The search performed by the swarm is to achieve the objective function which is either to be minimized or maximized. The personal best position (*pbest*) is obtained by evaluating the following.

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases} \tag{3}$$

The global best position (*gbest*) is obtained by using

$$\hat{y}(t) \in \{y_0, y_1, \ldots, y_s\} = \min\{f(y_0(t)), f(y_1(t)), \ldots, f(y_s(t))\} \tag{4}$$

The current position (x_i) and velocity (v_i) are recalculated after each iteration using

$$\begin{aligned} v_i(t+1) &= \omega v_i(t) + c_1 r_1(t)(y_i(t) - x_i) + c_2 r_2(t)(\hat{y}(t) - x_i(t)) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \tag{5}$$

where ω is the inertia weight which reflects the memory of previous velocities, thus controlling the influence of the previous velocity. The $y_i(t) - x_i$, cognitive component, represents the particle's own experience as to where the best solution is. The $\hat{y}(t)$–x_i , social component, represents the belief of the entire swarm as to where the best solution is. The c_1 and c_2 are acceleration constants and $r_1(t)$, $r_2(t) \sim U(0,1)$, where $U(0,1)$ is a random number between 0 and 1.

In image classification, the PSO algorithm is used to optimize the objective functions that are mainly to minimize the within cluster distance (between pixel and the cluster mean) and to maximize among cluster distance (between cluster pairs). In this algorithm, there is no known priori on the number of clusters. Hence the cluster validity verification is determined by the objective functions. The Davies-Bouldin index is used to evaluate the result of iteration [7].

5 The Proposed Rough K-Means Algorithm and Particle Swarm Optimization

The Rough K-means algorithm (hereafter is referred to as Rough K-means) is one of the methods to deal with vagueness of information. In the grey-scale image classification, the challenge is to segment the blurred boundaries between clusters. Using rough set theory, an image can be represented as sets of lower approximation and upper approximation. The Rough K-means model for our proposed image segmentation algorithm is adapted from [8].

$$c_j = \begin{cases} w_{lower} * \dfrac{\sum_{v \in \underline{A}(x)} v_j}{|\underline{A}(x)|} + w_{upper} * \dfrac{\sum_{v \in (\overline{A}(x) - \underline{A}(x))} v_j}{|\overline{A}(x) - \underline{A}(x)|}, & if \quad \overline{A}(x) - \underline{A}(x) \neq \varnothing \\ w_{lower} * \dfrac{\sum_{v \in \underline{A}(x)} v_j}{|\underline{A}(x)|}, otherwise \end{cases} \tag{6}$$

Each image pixel can actually be classified into lower or upper approximation. In the following the basic rough set properties are listed:

- A pixel can be a member of the lower approximation $\underline{A}(x)$
- If a pixel is part of the lower approximation $\underline{A}(x)$, then it is also part of the upper approximation $\overline{A}(x)$
- If a pixel does not belong to any lower approximation $\underline{A}(x)$, then it belongs to two or more upper approximations $\overline{A}(x)$.

Applying the rough set in the K-means requires the modeling of the algorithm to include lower and upper approximations. The equations (7) for the algorithm are shown below. Let v be a pixel vector and $d(v, c_i)$ be the distance between the pixel and the centroid of cluster C_i. Let

$$d(v, c_i) = \min_{1 \le j \le k} d(v, c_j) \tag{7}$$

$$T = \{ j : d(v, c_i) - d(v, c_j) \le threshold \quad and \; i \neq j \} \tag{8}$$

In order to classify a pixel to the right approximation(s), the following *classification criterias* are used:

- If the set, T, is not an empty set, then the pixel is classified as upper approximation of both cluster i and j.
- If T is an empty set, the pixel is classified as lower approximation for cluster i. Then the pixel is also classified as upper approximation for clusters i as well.

In summary, the following steps are required to perform the rough K-means algorithm.

1. Initialize N number of clusters randomly.
2. Select w_{lower} and threshold value.
3. For each cluster, find the d using equation (7) and T using equation (8).

4. Classify the pixel to lower and upper approximations using the *classification criteria*.
5. Calculate the new cluster center (mean) using equation (6).
6. If the algorithm converges, then stop. Otherwise, repeat steps 3 to 5.

The parameters involved are w_{lower}, w_{upper} and *threshold*. These parameters are set manually and required of the trial and error. It is not trivial to come up with good estimate values, and it is the major disadvantage for the rough K-means method. In order to determine these parameters automatically, this algorithm needs to be improved using the automatic tuning mechanism.

Rough K-means has two parameters that need to be determined, namely lower approximation variable (w_{lower}) and *threshold* value. Manual adjustment through testing is not practical for image segmentation in general. Hence, the particle swarm optimization is proposed to alleviate the limitation by automatically searching and modifying the parameters during the image segmentation process.

The proposed algorithm that combines Rough K-means and PSO algorithm is outlined as follows:

1. Initialize the cluster mean of each cluster c_k..
2. Initialize a number of particles p_i where each of the particles are randomly assigned with lower approximation variable (w_{lower}) and *threshold*.
3. Classify each pixel x in the particles to either lower or upper approximation of each cluster. Find the minimum pair of distance of x to all clusters, namely $d(x\text{-}c_i)$.
4. If the difference of the distance $d(x\text{-}c_i)$ and $d(x\text{-}c_j)$ is less than *threshold*, then the pixel belongs to upper approximation of both clusters c_i and c_j.
5. Otherwise, the pixel x belongs to lower approximation of cluster c_i.
6. Calculate the Davies-Bouldin Index (DB Index) of each particle. Each particle saves DB Index obtained from the iterations and compares them with the other particles'. Find the global best index and tune the lower approximation and thresholds of each particle according to the following guidelines.
7. If personal best DB Index = global best DB Index, then adjust *threshold lower* to include only the pixel that are definitely in lower approximation.
8. If personal best DB Index > global best DB Index, then adjust the *lower approximation variable* and *threshold* of the particle toward the values of particles with global best DB index.
9. Calculate the new mean for each cluster c_k.
10. Repeat steps 3 to 9 until all particles converge.

6 Experimental Results

To compare the effectiveness of these algorithms on segmenting images, preliminary experiments were conducted on several images. The experiments on the rough K-means are intended to show the effectiveness of parameters selection on the results of the classification. In Figure 1 (b), the parameter w_{lower} is 0.95 and the *threshold* is 0.05. The first parameter weights how much the previous calculated mean will affect

the new mean and the second parameter adjusts the boundary region of lower or upper bounds. In other words, the second parameter is the criteria limit whether a pixel shall be included as upper approximation of a class or not. The higher the value of the *threshold*, the more relaxed the criteria is. The test for the tool image shows a bit improvement on the far left screw image. The original image is shown in figure 1 (a). The K-means shows a bit distorted image, while the Rough K-means PSO shows much sharper screw image. Figure 2 shows the DB index values.

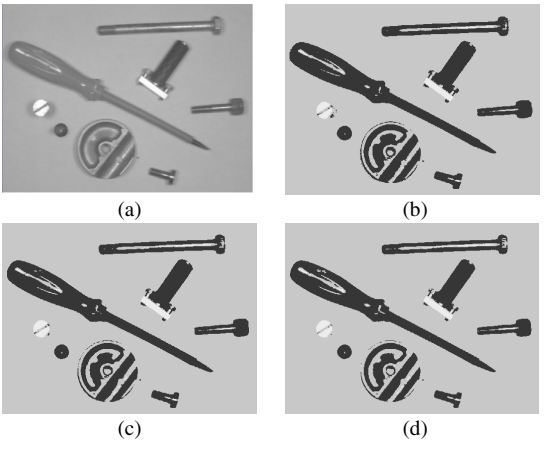


Fig. 1. Experimental results for tools image (a) Original (b) K-means (b) Rough K-means $w_{lower} = 0.95$ Threshold = 0.05 (d) Rough K-means PSO

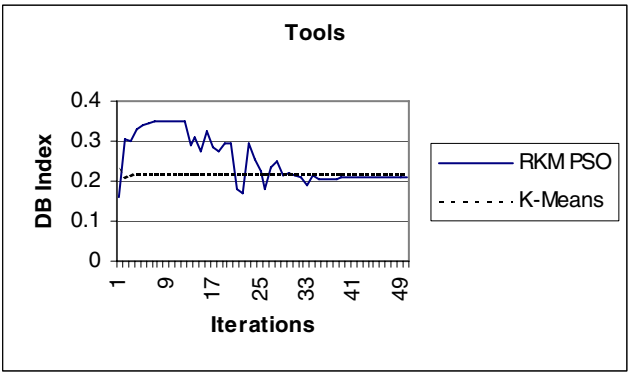


Fig. 2. K-means and Rough K-means PSO DB Index tracking sample for tools image

Testing for the tomcat image shows the rough K-means' performance. The original image is shown in Figure 3 (a). The tomcat jet image itself is much better shaped and defined. The K-means has difficulty in classifying the wing and cockpit area. The

rough K-means can better define the shape of the jet and discern the number 0001 on the jet bonet. This image is classified into five classes.

The overall result does not show too much improvement, as indicated by the DB index chart in Figure 4. However, blurred and vague areas can be classified well by the rough K-means PSO.

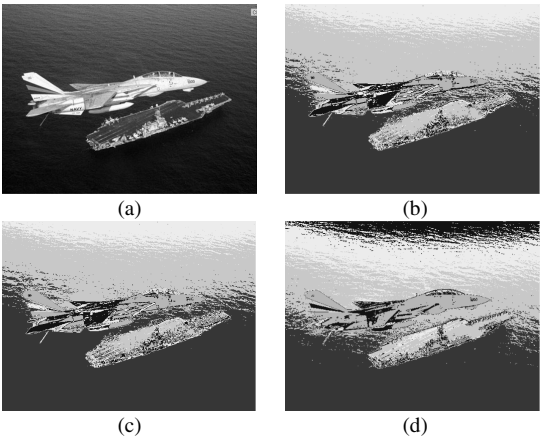


Figure 11.

Fig. 3. Experimental results for planet image (a) Original (b) K-means (c) Rough K-means $w_{lower} = 0.95$ Threshold = 0.05 (d) Rough K-means PSO

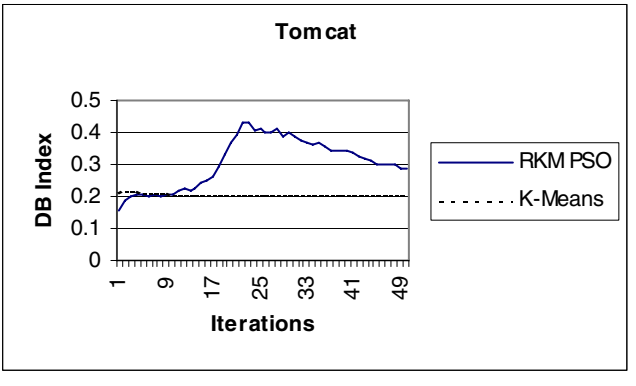


Fig. 4. K-means and Rough K-means PSO DB Index tracking sample for tomcat image

Finally, another test is performed using a gray scale aurora image as shown in Figure 5. Several tests are done for aurora images. The result shows that the Rough K-means PSO has better noise tolerance in the image. It appears to be able to reduce sporadic white spots around inner and outer aurora circle. The DB index values are shown in Figure 6.

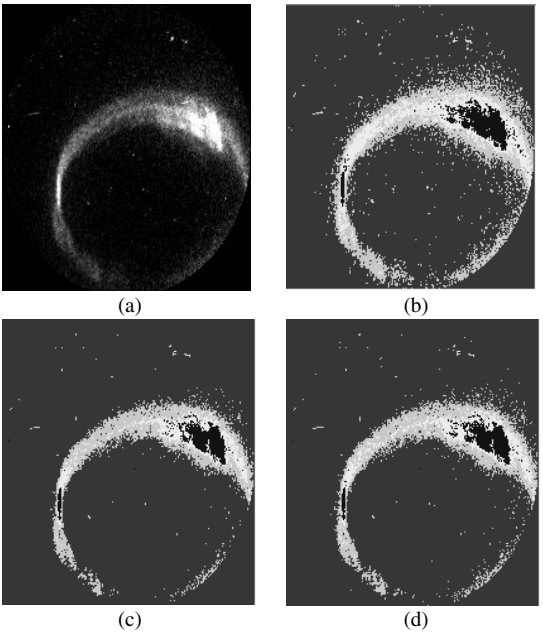


Fig. 5. Experimental results for planet image (a) Original (b) K-means (c) Rough K-means $w_{lower} = 0.95$ Threshold = 0.05 (d) Rough K-means PSO

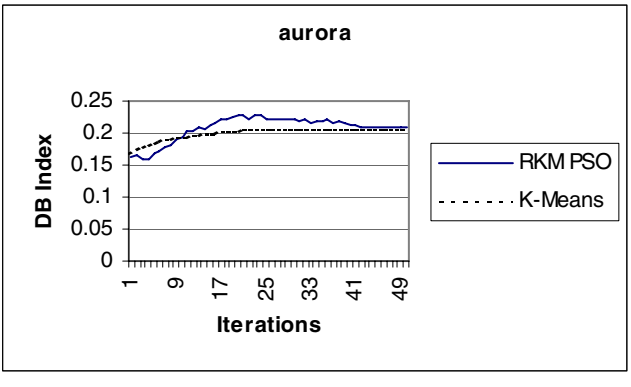


Fig. 6. K-means and Rough K-means PSO DB Index tracking sample for aurora image

6 Conclusions

The K-means algorithm can be improved using rough set theory. The Rough K-means algorithm has limited practical application, since the parameters (w_{lower} and *threshold*) are difficult to tune manually. The approach proposed in this paper is to employ PSO in tuning the parameters. The algorithm is tested on several images. In general, the proposed rough K-means PSO algorithm can improve the performance of the rough

K-means algorithm. The improvement can be seen in the experimental results, in particular, in the blurred and vague areas. The proposed algorithm also shows that it has better noise immunity in image segmentation as no pre-processing was applied to the image before the segmentation in the experiments.

Acknowledgement

This research was partially supported by NASA Science Mission Directorate grant NNG06GE60G.

References

[1] Komorowski, J., Pawlak, Z., Polowski, L., Skowron, A.: Rough Sets: A Tutorial. In: Pal, S.K., Skowron, A. (eds.) Rough Fuzzy Hybridization, pp. 3–98. Springer, Heidelberg (1999)
[2] Petrosino, A., Salvi, G.: Rough Fuzzy set based scale space transforms and their use in image analysis. International Journal of Approximate Reasoning 41, 212–228 (2006)
[3] Pawlak, Z.: Rough Set. Int. J. Inform. Comput. Sci. 11, 341–356 (1982)
[4] Tou, J., Gonzalez, R.: Pattern Recognition Principles. Addison-Wesley, Reading (1974)
[5] Hung, C., Germany, G.: K-Means and Iterative Selection Algorithms in Image Segmentation. In: IEEE SouthEast Conference, Session 1: Software Development, Jamaica, West Indies, April 4 - 6 (2003)
[6] Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, Perth (1995)
[7] Davis, D., Bouldin, D.: A Cluster Separation Measure. IEEE Trans. Pattern Analysis and Machine Intelligence 1, 224–227 (1979)
[8] Lingras, P., West, C.: Interval Set Clustering of Web Users with Rough K-means. Journal of Intelligent Information Systems 23(1), 5–16 (2004)
[9] Dariusz, M., Sławomir, T.: Standard and Genetic K-means Clustering Techniques in Image Segmentation. In: The 6th International Conference on Computer Information Systems and Industrial Management Applications, pp. 299–304 (2007)

Monocular Model-Based 3D Location for Autonomous Robots

Antonio Adán, Alberto Martín, Ricardo Chacón, and Vicente Dominguez

Escuela Superior de Informática, Universidad de Castilla La Mancha,
13071 Ciudad Real. Spain
Antonio.Adan@uclm.es, albmargue@yahoo.es,
Ricardo.Chacon@uclm.es, Vicente.Domingez@uclm.es

Abstract. The intelligence degree of a mobile robot is determined taking into account multitude of aspects concerning knowledge of the environment, intelligent decisions and suitable actions. The work presented in this paper focuses on answering an early and key question of an autonomous robot: where am I? We propose a monocular model-based strategy in which the eye of the robot can adapt focal and colour characteristics depending on the location of the robot. Through correspondences between a few points in the world coordinate system and their projections in the image plane of the camera, the method gives the 3D pose of the robot in the room and the location in the building. Our approach is being used for service robot applications inside buildings yielding excellent results. Experimentation, advantages and restriction of this technique are shown in the paper.

1 Related Works

In mobile robot navigation a key question that the robot frequently requires is "where am I?". This issue has been answered for years from many points of view using a wide variety of sensors (camera, sonar, infrared, laser,…) and technologies that yield information about the environment that surround the robot.

This paper deals with the problem of robot localization from a single landmark using a single measurement and a single camera. Throughout the paper we will frequently use the word pattern instead of landmark. This strategy is a minimum simplified version of the typical absolute method through triangulation of three or more measurements to different landmarks. Thus, the pose robot problem (position + orientation) is solved through the information extracted from a single image of a specific pattern. These approaches are included in monocular perspective projection techniques.

Stereo vision techniques have been widely experimented in indoor applications but require higher processing than monocular procedures [1]. Moreover both cameras must see the whole pattern to carry out image correspondence algorithms. This could be a serious constrain if the mobile robot runs in difficult environments like hospitals or museums where people became an obstacle. Consequently, monocular perspective projection techniques are suitable for on-line autonomous navigation because low processing is required [2]. Besides, occlusion problems are easily solved by taking another pattern in the same room.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 594–604, 2008.

Early works can be found in the literature from the 90's. Different patterns have been designed depending on the environments and accuracy required. Patterns composed of rectangles are used in [3] and more recently in [4]. Kabuba *et al.* [5] and Heikkilä [6] use projection of circular patterns.

Currently there are several approaches to pose estimation problem given some correspondences between 3D points in the world coordinate system and their projections in the image plane. Most of the authors have used correspondence between lines. Wang *et al.*. [7] extract information from patterns with two pairs of parallel lines which are non coplanar. Li-Juan *et al.* [8] propose an iterative method based on geometric relationships of three non-coplanar lines yielding very good results. A trapezium is used in [9] to estimate the camera pose. Other actual solutions using different discriminative landmark can be found in the literature (see references [11-13]).

Here we propose an original approach which allows us to control the pose of a mobile robot inside a building. We have designed an original landmark consisting of eight coloured spots corresponding to the vertices of an octahedron and one more double-size spot located at the centre of the octahedron.

Most techniques referenced above require precise image preprocessing stages to estimate the model from the pattern image. Thus, lines, vertices, circumferences, etc. have to be extracted in the image. As a consequence of this, faults and errors arise. If the volume of information to be extracted from the image grows and if the complexity of the features of the objects to be segmented augments the method may become inefficient. Lens distortion and digitization process contributes also to increase errors. In our case, the errors derived from the image processing decrease because we have just to extract nine spots in the image. Consequently, we avoid hard image processing (i. e. segmentation) and reduce highly the computational cost. Thus, the system can be considered 'on-line' for dynamic pose control in moving environments under minimum time requests.

The following sections give an overview of our approach. The camera model, the pattern and the pose parameters are dealt in section 2. Section 3 presents the robot self-location in a building whereas a summary of experimentation is reported in section 4. Finally section 5 deals with the conclusions.

2 Statement of the Problem and the Pattern

Let suppose a robot moving in a building. The absolute pose (position and orientation) of the robot, in a stated world coordinate system, is necessary information for autonomous navigation purposes. Since the relationships between the absolute coordinate systems inside the building are constant and known in advance, only the pose of the robot respect to the system defined by one specific landmark is necessary. We distinguish the following reference systems (see Figure 1 left): world reference system (S_w), robot reference system (S_2), camera reference system (S_1) and landmark reference system (S_0). Note that relationship S_w/S_0 is imposed when the landmark is put in a specific place and that S_1/S_2 is established in the robotic system itself. Therefore, the problem is reduced to find the transformation S_0/S_1. In summary, the absolute pose of the robot is calculated by means of $P_w=T_{w0}T_{01}T_{12}P_2$, where T_{w0} and T_{12} are known and fixed transformations. However, T_{01} varies as the robot or the cameras move.

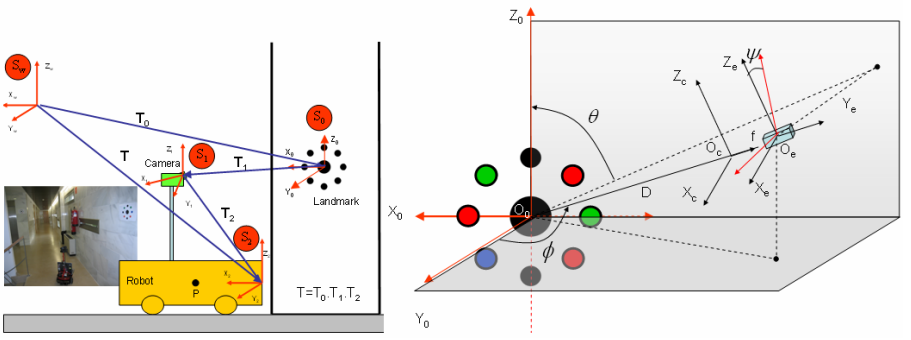


Fig. 1. Reference systems in autonomous navigation (left). Pattern and camera model (right).

We have designed an original landmark consisting of eight coloured spots corresponding to the vertices of an octahedron and one more double-size spot located at the centre of the octahedron. So, the reference system S_0 is centred at the center-point and the plane X_0Z_0 contains the rest of spots. Figure 1 right illustrates a generic pattern.

The procedure presented in this paper is based on the fact that for any view of the pattern the eight external points belong to an ellipse E which changes its parameters as the camera (or the robot) moves. Through geometric analysis of E and the location of the points on it, we are able to extract the angular parameters *swing* (ψ), *tilt* (ϕ), *pan* (θ) as well as the distance D' between the origin of S_0 and the camera.

2.1 Relationship between Pose and Ellipse's Parameters

Let (O_o X_o Y_o Z_o) be the pattern reference system (before called S_0) and let (O_e X_e Y_e Z_e) be the camera reference system. Origin O_e is located at the centre of the image plane – where the CCD matrix is placed- whereas Y_e is aligned with the optical axis. Assuming that the optical axis passes through O_o, we call O_c the optical centre, D the distance O_cO_o, f the focal of the camera and D' the distance O_eO_0.

The centres of the nine spots are labelled as $\{P_1,P_2,P_3,P_4,P_5,P_6,P_7,P_8,P_9\}$. Points P_1 to P_8 come from coloured spots whereas P_9 is the centre of a double-size black spot. Note that in the pattern, P_1 (*first point*) is on the Z_0 axis and P_9 and O_0 are coincident.

On the other hand, the computer reference system (O_s,X_s,Z_s) gives the coordinates in pixels, of the points in the digital image. We assume that pattern is centred in the image and, therefore, the optical axis is aligned with the line O_0O_e. As we will explain in section 3 this requirement is carried out in our system since a visual servoing is continuously executed until the pattern is centred in the image.

When the robot moves, a different aspect of the pattern is viewed. Next we will explain what is the consequence of the variation of parameters ψ, ϕ, θ and D' over the ellipse that adjusts the external points P_1 to P_8:

- Parameter ψ gives the rotation of the major axis of the ellipse in the image. Thus, a head on view verifies $\psi=0$ if axis O_eX_e is coplanary to the plane $Z_0=0$; otherwise the camera is rotated and $\psi\neq0$. Furthermore, for non frontal views $\psi\neq0$ as well.

- Parameter ϕ is defined as the angle defined by axis Y_0 and Y_e. If $\phi=0$, $\{P_1,P_2,P_3,P_4,P_5,P_6,P_7,P_8\}$ belong to a circle whereas if $\phi\neq0$ they belong to an ellipse.

- Parameter θ is the angle supported by Z_0 and the projection of the optical axis over the plane Y_0=0. When θ≠0, $\{P_1,P_3,P_5,P_7,\}$ are outside the axes of the ellipse.
- Parameter D' is the distance between O_e and O_0 and maintains a quasi-lineal relationship with major axis of the ellipse.

In order to obtain the complete location of the robot we follow two steps. Firstly, the robot-pattern pose is calculated. Then, we identify the pattern in the building and compute the absolute pose of the robot inside the building.

3 Pose of the Robot in a Building

3.1 Camera-Pattern Pose

In the next sections we present a novel 3D camera-landmark pose procedure based on the parameters of the ellipse that adjust the external points of the pattern in the image. From now on we will consider the image captured by the camera and the computer coordinates (X_s,Z_s) of the points $\{P_1,P_2,P_3,P_4,P_5,P_6,P_7,P_8,P_9\}$.

The general equation of the ellipse passing through $\{P_1,P_2,P_3,P_4,P_5,P_6,P_7,P_8\}$ is

$$Ax_i^2 + Bx_i z_i + Cz_i^2 + Dx_i + Ez_i = 1 \quad i = 1,2,...8 \tag{1}$$

where *A*, *B*, *C*, *D* and *E* are unknown in this equation. Eight linear equations with five unknowns are obtained from the eight points. Once this overdetermined system is solved, coefficients *A*, *B*, *C*, *D* and *E* are used to calculate *swing* and *tilt* angles.

As it said before, *swing* angle corresponds to the angle (ψ) between the major axis of the ellipse and the horizontal reference X_s. *Swing* angle ψ can be easily determined from the estimated parameters *A*, *B* and *C* from equation

$$\psi = \frac{1}{2} arctg \frac{B}{A-C}, A \neq C; \psi = 45^\circ, A = C \tag{2}$$

Tilt angle is obtained from eccentricity of the ellipse. In the next step we establish a set of geometrical relationships in which the distance camera-pattern D, the focal of the camera f, the major (a) and minor (b) axis appear. In order to understand the next equations Figure 2 shows the setup from a different viewpoint. Note that the following expressions are verified in this figure:

$$\frac{b}{m} = \frac{f}{D-n}; \begin{matrix} m = R\cos\phi \\ n = R\sin\phi \end{matrix} \tag{3}$$

$$b = \frac{fR\cos\phi}{D - R\sin\phi} \tag{4}$$

Assuming that $R.sin\phi << D$ (which is true in our case) equation (4) converts into

$$b = \frac{fR\cos\phi}{D} \tag{5}$$

On the other hand, from a perpendicular view to the major axis of the ellipse, the next relation is found

$$\frac{a}{R}=\frac{f}{D}\Rightarrow a=\frac{f}{D}R \tag{6}$$

Finally, from equations (5) and (6), we conclude that

$$\cos\phi=\frac{b}{a}\Rightarrow \sin\phi=e \tag{7}$$

e being is the ellipse eccentricity.

Therefore, we conclude that *tilt* angle is obtained from eccentricity of the ellipse fitted to the external points of the pattern. Values of ϕ are in the interval [0, 90º]. When $e=0$, $\phi=0$ and the points are fitted to a circle whereas when $e=1$, $\phi=90^{o}$ and the ellipse is converted into a segment.

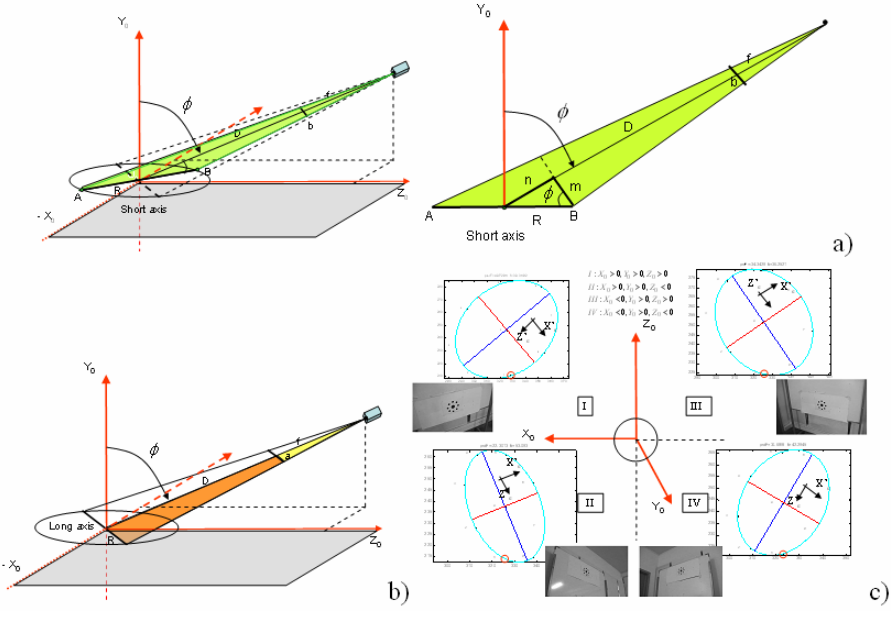


Fig. 2. Analysis of a circle projected in the image: minor axis (a) and major axis (b). Aspect of the fitted ellipse depending on the quadrant in which the camera captures the image (c).

Note that the major axis of the ellipse only depends on the distance from camera to P_9, as established in equation (6). Consequently the camera-pattern distance can be calculated through the focal and the major axis as follows:

$$D=\frac{f}{a}R \tag{8}$$

where a is the major axis in millimeters and corresponds to the projection of the pattern in the image plane. This expression proves that the camera-pattern does not depend on the rest of angular parameters.

Since we known the position of the CCD matrix in the camera, the distance D' can be measured experimentally. Therefore, in practice, we will use D' instead of D as camera viewing parameter.

Taking into account that in practice $D>>d$, d being the distance from the len's optical center to the image formation plane, f is a good approximation for d, and the next equation is verified.

$$D' \cong D + f \tag{9}$$

Thus, equations (8) and (9) can be used to estimate D' as follows:

$$D' \cong \frac{f}{a} R + f \tag{10}$$

Finally, *pan* parameter is obtained through the position of P_1 in the ellipse coordinate system. Let P be a point in the three-dimensional space. Let $P=P(x_e,y_e,z_e)$ and $P=P(x_0,y_0,z_0)$ be the image coordinates and the pattern coordinates of P. The transformation between S_0 and S_e reference systems can be calculated by means of concatenate elemental transformations as follows:

$$T = C.R_{Y_0''}(\pi/2).R_{X_0'}(\phi).R_{Y_0}(-\theta) \tag{11}$$

Where the Euler rotations $R_{Y_0''}(\pi/2), R_{X_0'}(\phi), R_{Y_0}(-\theta)$ are performed over the rotated axis after the first and second transformation and matrix C corresponds to a translation in axis Y_0''.

Assuming that $\psi=0$, the transformation of the systems S_0 and S_e is given by:

$$\begin{pmatrix} x_e & y_e & z_e & 1 \end{pmatrix} = . \begin{pmatrix} \sin\theta & 0 & \cos\theta & 0 \\ \sin\phi\cos\theta & \cos\phi & -\sin\phi\sin\theta & D \\ -\cos\phi\cos\theta & \sin\phi & \cos\phi\sin\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{pmatrix} \tag{12}$$

For $P=P_1$, $P_1 = (0,0,-R,1)_{S_0}$, equation (12) yields $x_{e1} = -R\cos\theta$ and $z_{e1} = -R\cos\phi\sin\theta$. Then $tg\theta = \frac{z_{e1}}{x_{e1}\cos\phi}$ and taking into account that $\frac{x_e}{z_e} = \frac{x_s}{z_s}$ we finally obtain:

$$tg\theta = \frac{z_{s1}}{x_{s1}\cos\phi} \tag{13}$$

In the case $\psi \neq 0$, unrotated coordinates x'_s and z'_s must be previously obtained and introduced in last equations. It can be seen that pan angle estimation depends explicitly on *tilt* and *swing* angles. Therefore, errors produced in the estimation of these parameters are propagated to θ.

Equation (13) yields indeterminate values of θ. This problem can be resolved by knowing the quadrant (in the system S_0), where the camera is placed. This quadrant is established through the sign of ψ and the position of the point P_9 in the minor axis. It can be proved that, due to projective reasons, P_9 is displaced with respect to the

theoretical ellipse center. As a consequence of this, distances d_1 and d_2 from P_9 to the ellipse, in the minor axis direction, are different. Therefore, we can infer whether the pattern is viewed from the left (case $d_1 > d_2$) or from the right (case $d_1 < d_2$). Figure 2 c) illustrates the definition of the quadrants and presents the displacement of P_9.

3.2 Pattern Identification in the Building

In order to identify each pattern, we have introduced a colour code. Distribution of colours to points can be easily changed according to the building considered. In order to prevent errors or small displacement in the computer coordinates of points P_1 and P_9 - which are essential in the approach -, we have maintained black colour for their corresponding spots. The rest of the spots can be red, green or blue. Black colour can be also painted in P_3, P_5 and P_7. Information included in the pattern is as follows:

- Floor code (f). It identifies the floor where the pattern is. In our case we have chosen only the point P_5 (four colours) which gives up to 4 floor codes. It is enough for our purpose but more or less points could be taken in the case of buildings with more or less floors.
- Room code (r). Colours assigned to P_6, P_7 and P_8 identify the number of room on each floor. This could code up to 36 rooms/floor.
- Pattern code (p). It identifies the number of pattern in each room. This means that it could have more than one pattern inside every room. For this, we have chosen points P_2, P_3 and P_4. So we have 36 patterns/room.

In short, to determine the position of the robot inside the building we could consider six coordinates: (f, r, p, x_0, y_0, z_0). Where (x_0, y_0, z_0) are coordinates in pattern coordinate reference system S_0. As it is clear, transformations between all pattern coordinate systems are known a priori. Therefore a building coordinate system (O_w X_w Y_w Z_w) could be defined and (x_w, y_w, z_w) would provide absolute position of the robot.

In our case, we have considered an absolute reference system by room (O_r X_r Y_r Z_r) which is fixed on one specific pattern. Then we pass (x_0, y_0, z_0) to (x_r, y_r, z_r) and finally absolute location of the robot is provided by (f, r, p, x_w, y_w, z_w).

4 Experimental Results

The proposed method has been implemented on Pioneer I multisensor mobile robot. In order to prove the applicability of our method under different environments we have tested the pose algorithm on two cameras of different characteristics. KC-N600PH CENTER Camera is a B/W auto-iris camera focused on security applications what can work in light/dark conditions whereas SONY EVI D-31 camera is a controlled azimut/latitude color camera with auto-focus and controlled zoom on board.

When the robot requires its pose, the followings phases are executed on board: 1) find the pattern, 2) center and maximize the pattern in the image, 3) find the spots centers 4) calculate parameters ψ, ϕ, θ, D' as well as the whole transformation T, 5) identify the pattern in the building through its colour code. Figure 3 shows the steps using the monochrome KC-N600PH camera.

Color pattern has been designed with a concentric set of spots 127mm (P_1 to P_8) and 254 mm (P_9) diameter. On the other hand, Black-R-G-B color coding has been robust enough for our diffuse lighting environments. The colour code was defined in section 3.

We have carried out three types of tests: the robot moving around only one landmark, the robot moving in a room and the robot moving in a building.

In the frist experiment we measure the true position of the robot using a Leica DISTO^{TM} A6 laser tape measure, where the data can be electronically transferred to a pocket PC or Lapt. It allows measuring of long distances (up to 200m) with accuracy ± 1.5 mm.

We took 80 positions with KC-N600PH CENTER Camera and 45 positions with SONY EVI D-31 camera. The system worked in all cases although the quadrant was wrong once in the second experiment. Average absolute error values for both cameras were: *e(D)=1,32cm, e(X)=2,43cm, e(Y)=1,00cm, e(Z)=1,67cm* and *e(D)=2,14, e(X)=2,80cm, e(Y)=4,10cm, e(Z)=2,92cm*. Table 1 summarizes statistical results of the errors obtained for each estimated robot's coordinate and robot-pattern distance for both cameras. Absolute and relative errors are presented in two tables. For each case, average, standard deviation, greatest and smallest errors are presented as well.

Table 1. Results of the Test No 1

KC-N600PH CENTER camera

Abs. Errors (cm)	**e(X)**	**e(Y)**	**e(Z)**	**e(D')**
Average.	2,43	1,00	2,05	1,32
Std.Dev	1,44	0,75	4,15	0,94
Greatest	5,41	3,10	6,02	3,49
Smallest	0,11	0,04	0,25	0,01
R. Errors (%)	**e(X)**	**e(Y)**	**e(Z)**	**e(D')**
Average	5,23	0,83	3,25	0,83
Std.Dev	6,68	0,52	4,02	0,48
Greatest	31,93	2,20	6,03	1,83
Smallest	0,30	0,01	0,95	0,02

SONY EVI D-31 camera

Abs. Errors (cm)	**e(X)**	**e(Y)**	**e(Z)**	**e(D')**
Average,	2,80	4,10	2,92	2,14
Std.Dev	2,06	2,01	1,73	2,11
Greatest	8,34	7,13	6,53	9,19
Smallest	0,01	0,11	0,07	0,07
R. Errors (%)	**e(X)**	**e(Y)**	**e(Z)**	**e(D')**
Average	3,58	3,08	6,87	1,41
Std.Dev	3,26	1,72	5,06	1,47
Greatest	14,10	6,81	31,47	4,85
Smallest	0,00	0,07	0,38	0,03

In the second experiment the robot moved in two labs of 70 and 71 m^2 where there were a lot of obstacles (tables, chairs, bookcases and also persons). We tested different options for each one of the labs. In order to the robot was able to see at least one landmark from any position, we made an experience putting seven identical black landmarks and put them inside the lab. In this experiment we did not used a coloured code because we wanted to know whether the system was able to identify each pattern by means of the odometry of the robot. In general we concluded that although the odometry yields considerable errors the system work for minor walks. Therefore, the robot followed a circuit of 42 m length around the lab avoiding obstacles by means the ultrasonic sensors. In the second option a coloured pattern was used maintaining P_1, P_3, P_5, P_7 and P_9 in black. As in first test, we found small positioning error always less to 5 cm in both experiments. Shorter and longer distance from the robot to the

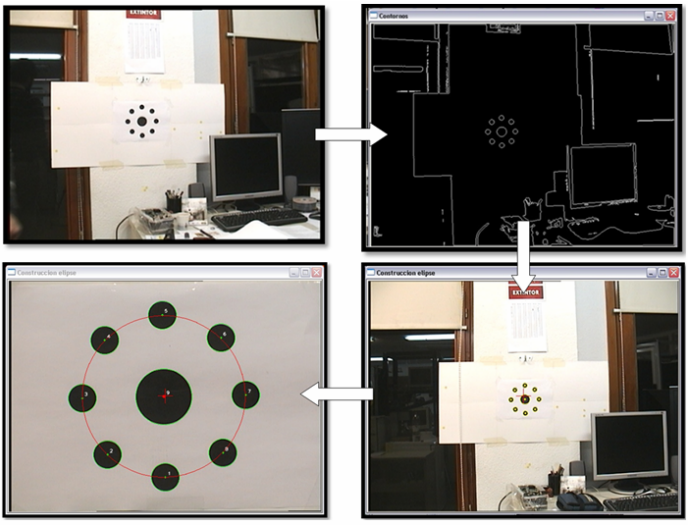


Fig. 3. Detecting and maximizing the pattern in the center of the image

Table 2. Results of the Test No 2

LAB I

Abs. Errors cm	**e(X)**	**e(Y)**	**e(Z)**	**e(D')**
Average.	2.55	2.26	3.20	2.52
Std.Dev	2,07	3,41	3,20	2,12
Greatest	7,86	9,06	9,02	7,36
Smallest	0,54	0,00	0,32	0,68
R. Errors (%)	**e(X)**	**e(Y)**	**e(Z)**	**e(D')**
Average	1,08	0,80	4,53	1,24
Std.Dev	1,08	1,73	4,58	0,84
Greatest	4,02	5,31	12,8	2,55
Smallest	0,09	0,00	0,47	0,27

LAB 2

Abs. Errors cm	**e(X)**	**e(Y)**	**e(Z)**	**e(D')**
Average	2,47	2,33	4,76	2,60
Std.Dev	3,18	1,71	4,51	1,96
Greatest	10,11	6,53	13,69	5,66
Smallest	0,18	0,30	0,14	0,27
R. Errors (%)	**e(X)**	**e(Y)**	**e(Z)**	**e(D')**
Average	0,60	0,64	6,53	1,52
Std.Dev	0,51	0,45	6,49	1,56
Greatest	1,76	1,42	20,80	5,69
Smallest	0,03	0,06	0,17	0,11

recognized pattern were 117,6 cm and 366,5 cm for the first and 88,9cm and 284 cm for the second one. We also present a summary of the results in table 2.

The last experimental report concerns to the autonomous movement of the robot in a more extend surface. The building where we have tested our method has 2 floors with room dimensions from 16m^2 (small offices) to of 70m^2 (labs). The building includes a pair of corridors, lift, stairs and two baths per floor. To start with, the path planning is introduced in the robot computer stating the start and final position as well as a set of intermediate points along the trajectory. The robot checks its pose applying the procedure presented in this paper in regular time intervals and recalculates the trajectory. Figure 4 shows the building and the path designed. Also we present images

Fig. 4. Above: Building and paths of the robot in he first and second floor. Below: Different locations of the robot in the route and patterns founded.

of the robot in some of the critical points. For instance we can see the mobile robot taking the lift to go the second floor.

The mobile robot completed successfully the route without major problems and always was able to calculate the pose choosing the best visible landmark. A wider experimentation with more difficult and long robot routes is being accomplished in this environment. Also we are currently dealing cases in which the landmark appears partially occluded. For the moment the algorithm is being modified in order to be robust enough under these circumstances. Our goal is to achieve an efficient and robust pose system to be applied in service-robot applications. We aim to implant this technique for guide-robots in museums and for vigilance-robots in parking.

5 Conclusions

The method presented in this paper solves the robot 3D location problem using a single color camera on board a mobile robot. Therefore it is simpler than stereoscopic methods or triangulation based methods where more than one image is used. The pose is calculated after analyzing the projected image of an artificial landmark consisting of nine colored spots.

3D location involves two things. Firstly, the robot is not restricted to run over a flat floor as in 2D navigation. Secondly, a color code, containing information about

location of each room, has been inserted in the pattern. Therefore a full location of a mobile robot can be obtained when the camera looks at the pattern.

Due to the simplicity of the image processing our method is fast and robust. Therefore it is suitable for on-line robot location applications. However, it must be mentioned that the centre of the landmark must be centered in the image.

Our approach is being used for applications inside buildings yielding excellent results. Experimentation with a mobile robot, advantages and restriction of this technique have been shown in the paper.

References

1. Kriegman, D.J., Triendl, E., Binford, T.O.: Stereo vision and navigation in buildings for mobile robots. IEEE Trans. Robot. Automat. 5, 792–803 (1989)
2. Guiducci, A.: Camera calibration for Road Applications. Comp. Vision and Image Understanding 79, 250–266 (2000)
3. Haralick, R.M.: Determining camera parameters from the perspective projection of a rectangle. Pattern Recognition 22(3), 225–230 (1989)
4. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trnasactions on Pattern Analysis and machine Intelligence 22(11), 1330–1334 (2000)
5. Kabuka, M.R., Arenas, A.E.: Position verification of a mobile robot using standard pattern. IEEE Journal of robotics and automation 3(6) (1987)
6. Heikkilä, J.: Geometric Camera Calibration Using Circular Control Points. IEEE Trans. PAMI 22(10), 1066–1077 (2000)
7. Wang, G., Wu, J., Ji, Z.: Single based pose estimation from circle or parallel lines. Pattern Recognition Letters 29, 977–985 (2008)
8. Li-Juan, Q., Feng, Z.: A new method for pose estimation from line correspondences. Acta Automatica Sinica 34, 130–134 (2008)
9. Duan, F., Wu, F., Hu, Z.: Pose determination and plane measurement using a trapezium. Pattern Recognition Letters 29, 223–231 (2008)
10. Nguyen, X., You, B., Oh, S.: A simple landmark model for vision-based simultaneous localization and mapping. In: Intern. Joint Conference SICE-ICASE 2006, pp. 5016–5021 (2006)
11. Zitová, B., Flusser, J.: Landmark recognition using invariant features. Pattern Recognition Letters 20, 541–547 (1999)
12. Guo, Y., Xu, X.: Landmark design using projective invariant for mobile robot localization. In: International conference on Robotics and Biomimetics, pp. 852–857 (2006)
13. Ikeda, S., Sato, T., Yamaguchi, K., Yokoya, N.: Construction of feature landmark database using omnidirectional videos and GPS positions. In: Sixth international conference on 3-D digital imaging and modeling (2007)

A Robust Iterative Closest Point Algorithm with Augmented Features

Carlos Lara, Leonardo Romero, and Félix Calderón

Universidad Michoacana de San Nicolás de Hidalgo
División de Estudios de Posgrado, Facultad de Ingeniería Eléctrica
Santiago Tapia 403 Centro, Morelia, Michoacán, México, CP 58000

Abstract. The Iterative Closest Point (ICP) is widely used for 2D - 3D alignment when an initial estimate of the relative pose is known. Many ICP variants have been proposed, affecting all phases of the algorithm from the point selection and matching to the minimization strategy.

This paper presents a method for 2D laser scan matching that modifies the matching phase. In the first stage of the matching phase our method follows the ordinary association strategy: for each point of the new-scan it finds the closest point in the reference-scan. In a second stage, the most probable normal vector difference is calculated and associations that do not fulfill the normal vector difference requirement are re-associated by finding a better association in the neighborhood. This matching strategy improves the ICP performance specially when the initial estimate is not close to the right one, as it is shown in the simulated and real tests.

Keywords: Mobile Robotics, ICP, Scan Matching.

1 Introduction

A large number of today's mobile robots use a 2D laser range finder as a proximity sensor because it is fast and accurate. Laser Scans are collections of points that can be used directly to perform simple tasks. For high level tasks, such as Localization and Mapping, more processing is necessary (e.g. to build a map of an environment). Most of these high level tasks need to track the position of the robot.

The aim of the *Scan Matching* (SM) techniques is to estimate the relative motion $T = [\Delta x, \Delta y, \Delta\theta]^T$ (translations in both axes and rotation) of a robot given two consecutive scans of m laser measurements, $Z_{ref} = \{p_1, ..., p_m\}$ and $Z_{new} = \{q_1, ..., q_m\}$, called the reference scan and the new scan respectively. Scan matching approaches can be categorized into three groups depending on the type of association they use: a)Point to Point, b) Feature to Feature, or c)Point to Feature approaches [6]. The main advantage of *Point to Point* approaches is that they do not require to find features in the environment, specially when features are hard to find.

The Iterative Closest Point (ICP) algorithm, developed by Besl and McKay [1] and Chen and Medioni [3], is the most popular point to point approach.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 605–614, 2008.

Algorithm 1. ICP

1: Let be $Z_{ref} = \{p_1, ..., p_m\}$ and $Z_{new} = \{q_1, ..., q_m\}$ two scans, and be $T_0 = [\Delta x_0, \Delta y_0, \Delta\theta_0]^T$ an initial pose estimation,
2: Transform Z_{new} using T_0
3: $T \leftarrow T_0$
4: **repeat**
5: For each p_i in Z_{ref} compute the closest point in Z_{new} obtaining a set of correspondences $C = \{(p_i, q_k)\}$.
6: Find $T' = [\Delta x', \Delta y', \Delta\theta']^T$ such that minimize $E = \sum_{\forall i} [d(p_i, q_k)]^2$
7: Transform Z_{new} using T' and update T
8: **until** Convergence
9: T is the best rigid transformation

This paper proposes a new version of the ICP that modifies the matching phase by selecting the closest point that is compatible with the Most Probable Angle (MPA). This new approach, described below, helps specially when the initial estimation is far from the right one and the ICP is not able to find the right position.

This paper is organized as follows: Section 2 describes relevant related works around the ICP Method, Section 3 describes the new version of the ICP, Section 4 shows some experimental results and Section 5 presents our conclusions.

2 Related Work

Algorithm 1 shows the ICP method: given two sets of partially overlapping points, Z_{ref} and Z_{new}, and one initial estimation of their relative positions, T_0, the ICP method finds the rigid transformation, T, between the two scans in an iterative way. At each iteration the ICP method considers that each point p_i of the first scan corresponds to its closest point q_k in the second scan. After that ICP calculates the relative position that minimizes a metric distance, d, between points p_i and q_k. The algorithm guarantees that a local minimum is reached and it is not always the desired result.

ICP success depends on a good initial approximation, this assumption is usually fulfilled in robotics when there is a good enough displacement estimation given by odometry. However, sometimes, ICP can reach only a local minimum, which may be far from the overall minimum. Figure 1 shows an example of a wrong association of points (small circles in the Figure) and then the ICP method is not able to find the right transformation to match both scans. Because most of the red points have a blue point in the neighborhood (for the initial position shown in the Figure), points in the corners of the red square do not help to change the result of the ICP (illustrated by the Figure). Lines in the Figure helps to see that a 45° rotation is needed to match the red scan to the blue one.

A crucial decision in the ICP algorithm is how to calculate the correspondence between points. The closest point association criterion is not always the best decision. Rusinkiewicz [12] studies some strategies to find better correspondences.

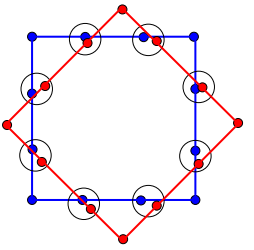


Fig. 1. The ICP method depends on the initial estimation T_0. In this case, ICP fails to match all the points.

Other authors have proposed to use features to improve ICP algorithm. Features are recognizable structures of objects in the environment. One idea is to select the closest point that is compatible with the source point according to a selected feature. Chua and Jarvis [4] use principal curvatures to find 3–tuples that guide the search. Sharp et. al. [13] use shape features (curvature, moment invariants, spherical harmonics invariants) in conjunction with point positions; point correspondences are chosen as the closest point with respect to a weighted linear combination of positional and feature distances. Lu and Milios [8] introduce the Iterative Dual Correspondence (IDC) algorithm. The IDC algorithm is an ICP variant that solves the point correspondence problem based on two heuristics: the closest point rule and the matching range rule. The latter rule chooses another correspondence point as the model point which has the same distance from the origin as the data point.

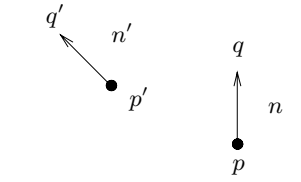
Normal (or tanget) vectors has been used by several authors to improve ICP. Zhang [14] imposes a simple constraint: the angle between tangents of two paired points should not be bigger than a prefixed value; in a similar manner, Pulli [11] allows points to match only if their associated normal vectors differ by less than 45 degrees. Lu and Milios [8] use a fixed threshold to reject bad correspondences based on tangent vectors.

This paper proposes a new version of ICP that uses normal vectors as features. The main difference between our method and the approaches of Zhang, and Pulli is the use of the Most Probable Angle (MPA) instead of a prefixed value. The MPA improves the point correspondence and it is a key idea of this paper.

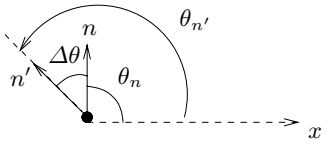
3 Scan Matching Algorithm

Consider that a point p and its associated normal vector n are transformed using $T = [\Delta x, \Delta y, \Delta\theta]^T$ to get $p' = Rp + t$ and n' as it is shown in figure 2(a). R is a rotation matrix associated to $\Delta\theta$ and t is a translation vector associated to Δx and Δy. Considering points as vectors, the final points are $q = p + n$ and $q' = p' + n'$ respectively; and hence:

$$\begin{aligned} n' &= q' - p' = (Rq + t) - (Rp + t) \\ &= Rq - Rp \\ &= Rn \end{aligned}$$



(a) Transformation of point p and normal n into p' and n'



(b) Relations between angles

Fig. 2. Normal vector relationship

this means that normal n' is independent of translation. In other words, normal n is rotated by $\Delta\theta$ to get n'. Figure 2(b) illustrates this relation. The angle between normals can be calculated as:

$$\Delta\theta = \theta_{n'} - \theta_n \tag{1}$$

Suppose that we have two point sets $Z_{ref} = \{p_i\}$, $Z_{new} = \{q_j\}$ related by a rigid transformation and the true correspondence $C = \{(p_i, q_k)\}$ is known. In an ideal situation, each point p_i in the first scan is related to the point q_k by $Rp_i + t = q_k$, therefore $\Delta\theta_i = k$, $\forall i$. In real situations, noisy measurements cause some deviation in the $\Delta\theta_i$ value.

The proposed algorithm is shown in Algorithm 2. It adds two steps to the basic ICP: step 6 estimates the most probable angle $\Delta\theta$ and its standard deviation σ; and, step 7 improves the correspondence C using $\Delta\theta$. Next sections describe the implemented algorithm.

3.1 Matching Heuristics

Finding point correspondences between maps is an important step in the ICP. The matching phase of the algorithm 2 uses a two stages strategy. In the first stage (Step 5) our method follows the ordinary association strategy: for each point of the reference-scan it finds the closest point in the new-scan. For laser scans, the closest point can be located by simply matching points with the same bearing [5]. In the general situation, finding the closest point is computationally expensive, and structures such as kd-tree [7] are useful to improve speed.

In step 6 the Most Probable Angle (MPA), $\Delta\theta$, is calculated. Let $C = \{(p_i, q_k)\}$ be a given correspondence between set points calculated by the closest point rule, and $\{\Delta\theta_i\}$ be the set of differences between their vector normals. The

Algorithm 2. The new version of the ICP

1: Let be $Z_{ref} = \{p_i \mid i = 1 \dots m\}$ and $Z_{new} = \{q_j \mid j = 1 \dots m\}$ two scans, and be $T_0 = [\Delta x_0, \Delta y_0, \Delta\theta_0]^T$ an initial pose estimation,
2: Transform Z_{new} using T_0
3: $T \leftarrow T_0$
4: **repeat**
5: For each $p_i \in Z_{ref}$ compute the closest point in $p_j \in Z_{new}$ obtaining a set of correspondences $C = \{(p_i, q_k)\}$.
6: Find the most probable angle $\Delta\theta$ and its standard deviation σ
7: Based on C get a new set of correspondences $C' = \{(p_i, q_l)\}$ such that $|d(p_i, q_l)| < \delta$ and $\Delta\theta - \sigma < \Delta\theta(p_i, q_l) < \Delta\theta + \sigma$
8: Find $T' = [\Delta x', \Delta y', \Delta\theta']^T$ such that minimize $E = \sum_{\forall i} [d(p_i, q_l)]^2$
9: Transform Z_{new} using T' and update T
10: **until** Convergence
11: T is the best rigid transformation

MPA is the most probable value for $\Delta\theta_i$ and this value (a rotation for all the points) is used to get a better point association. MPA can be approximated by the media of $\{\Delta\theta_i\}$.

In a second stage (Step 7) of the matching algorithm, correspondences that do not fulfill MPA requirement are re-associated by finding a compatible association in the neighborhood.

This matching strategy improves the ICP performance specially when the initial estimate is not close to the right one. Due to their importance, our algorithm uses a robust estimation of the normal vector to each point.

3.2 Error Metric

Instead of minimize the distance between two corresponding points[1], Chen and Medioni[3] minimizes the distance between one of the points and the plane passing through the other. This metric is known as Point–to–Plane distance (3D) or Point–to–Line distance (2D), and it can be stated as:

$$E = \sum_{\forall i} \left[(Rp_i + t - q_i) \cdot n_i \right]^2 \tag{2}$$

where R is a Rotation matrix, t is a translation vector and n_i is the normal of the $i-$th point of the reference scan. Other metrics also have been used successfully [9].

To find the best transformation, the steepest descendent method [10] is used with $\Delta\theta$ as the starting position of the search.

3.3 Normal Vector Estimation

Calculating normal vectors can be done by simply estimating the Tangent Line at each point, thus the orthogonal unitary vector to the tangent line is an approximation of the normal at such point. The tangent line at point p_i is often

calculated by fitting a line to the set of k nearest neighbors (in the neighborhood) of the point p_i. This approach works well in the presence of smooth regions and moderate noise, but it performs poorly in those regions near corners. If the neighborhood at each point has a fixed size and it is constructed using only the Euclidean distance then it is possible that points considered as outliers have a huge impact in the computation of the normal.

A Robust Normal Estimation can be performed using the NENR algorithm [2]. This algorithm produces a reduction in the neighborhood size, rejecting neighbors that have large normal vector differences. This condition warranties that the neighbors have the same smoothness degree between them.

4 Experimental Results

In order to test proposed algorithm, we carried out simulated and real tests. Proposed algorithm results were compared with ICP results. For both cases, a steepest descendent method was implemented to get the best transformation.

4.1 Simulated Data

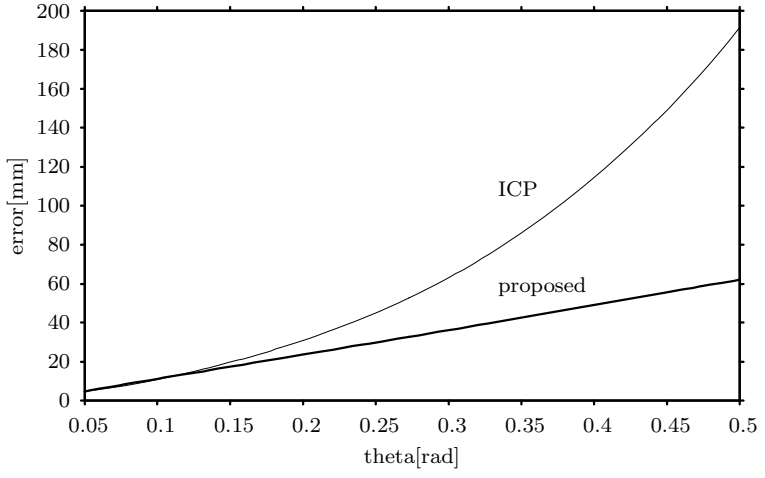
Simulated data is useful because the correct rigid transformation is known. A laser scan that covers 180° with an angular resolution of 0.5° was simulated adding gaussian noise with $\sigma = 40$ mm. To evaluate how close is the solution found by each method an error value e is computed as follows:

$$e = \sqrt{e_x^2 + e_y^2 + a e_\theta^2}$$

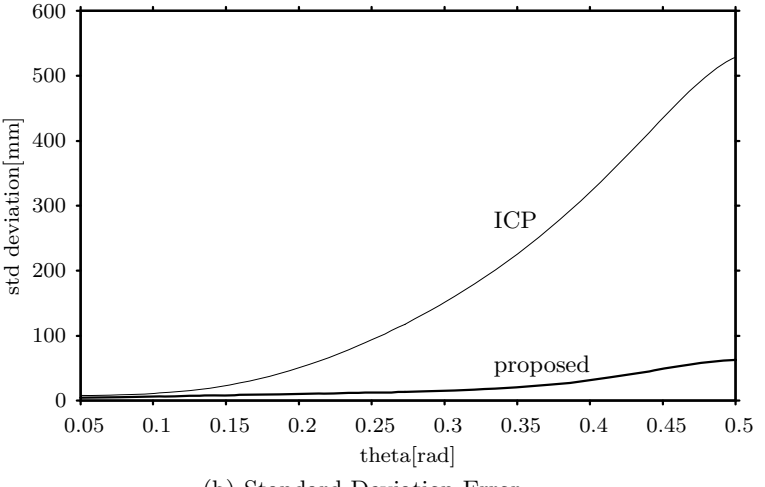
where e_x and e_y are translational errors, e_θ is the rotational error and $a = 1 \times 10^6$. Figure 3 shows results for a test on which 100 different simulated environments were created, translation error was fixed to $\Delta x = \Delta y = 100$mm, and rotation error was varied in a range from 0 to 0.5 rads. Figure 3(a) shows that for small rotation errors ICP and the proposed algorithm have almost equal behavior. For larger rotation error our algorithm shows a lower error. On other hand, Figure 3(b) shows that standard deviation is always better for the proposed algorithm. A method with small variance is important in many tasks specially when the robot is building a map of the environment and it needs to know its location with high accuracy.

4.2 Real Data

The algorithm described above was tested in a set of range images taken by our real mobile robot equipped with a LMS209–S02 SICK Laser Measurement System. Each laser scan covers 180° with a lateral resolution of 0.5°, a total of 361 points per laser scan. Figure 4(a) shows an example of a wrong association obtained by using simple closest point criterion (see associations of points marked



(a) Average Error

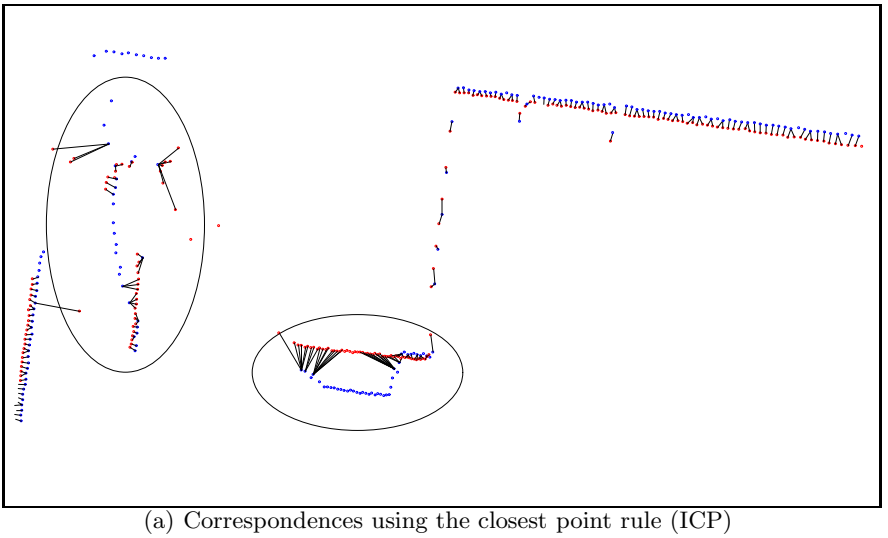


(b) Standard Deviation Error

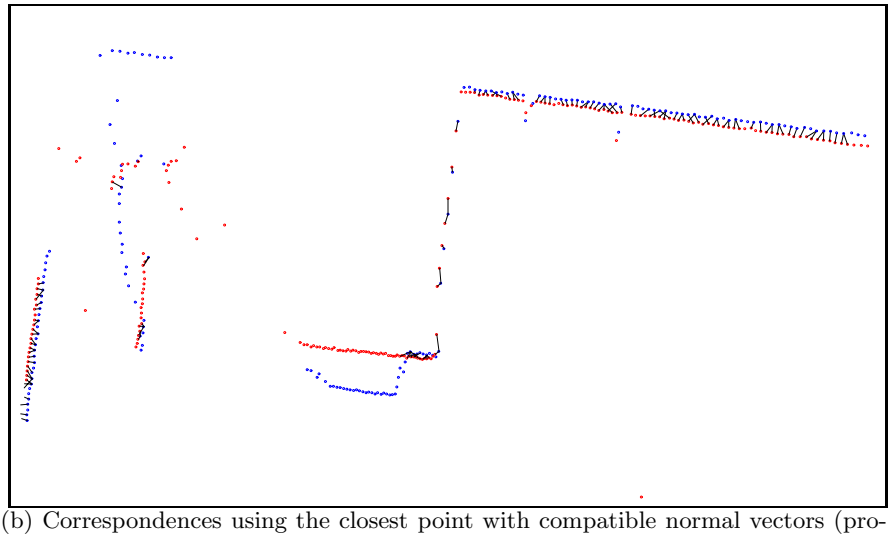
Fig. 3. Results of simulated data. Results of the proposed approach are better than results of the ICP.

by ellipses). Figure 4(b) shows a better point correspondence by considering the most probable angle. In both figures point correspondences are denoted by lines that join related points.

Figure 5 shows the final results of both methods. The reference scan is plotted as well as the new scan using the transformation found by each method. Point correspondences can modify substantially the result, the incorrect estimation of the rigid transformation shown in Figure 5(a) is due to wrong point associations shown in Figure 4(a). Figure 5(b) shows that the proposed algorithm finds a better estimation.

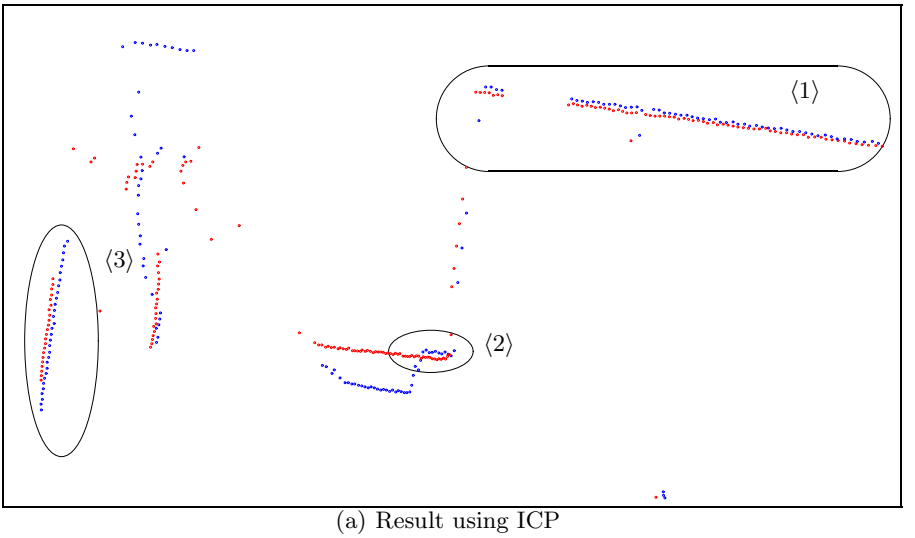


(a) Correspondences using the closest point rule (ICP)

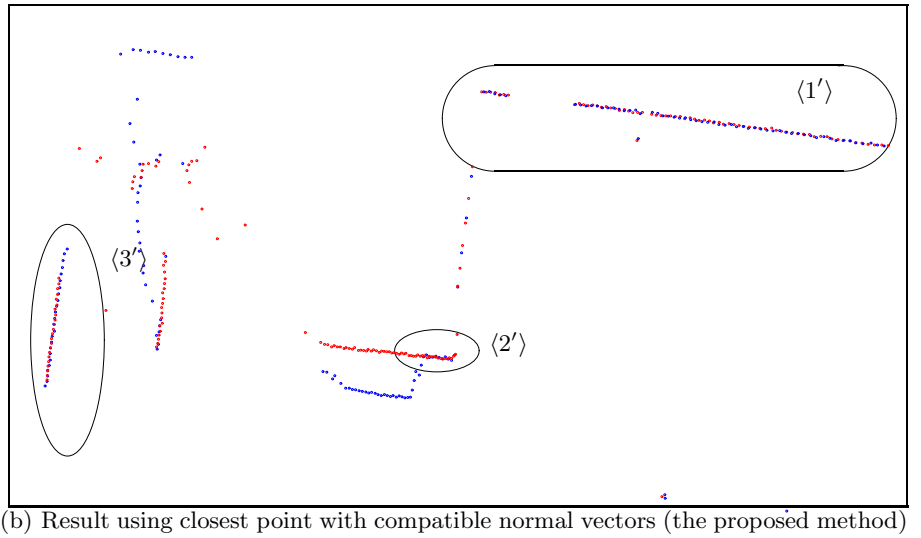


(b) Correspondences using the closest point with compatible normal vectors (proposed method)

Fig. 4. Point correspondences. The reference-scan is represented by red points while the new-scan is denoted by blue points, a line joining a blue point and a red point represents a single point association. As can be seen, closest point strategy gives some wrong associations (marked out by ellipses). The maximum distance allowed for point association is 30mm.



(a) Result using ICP



(b) Result using closest point with compatible normal vectors (the proposed method)

Fig. 5. Final result. **5(a)** ICP Estimation has some errors (marked out by ellipses ⟨1⟩, ⟨2⟩ and ⟨3⟩). **5(b)** A better estimation is obtained by the proposed algorithm (marked out by ellipses ⟨1′⟩, ⟨2′⟩ and ⟨3′⟩).

5 Conclusions

Local localization is one of the most important tasks for mobile robots. One way to solve the local localization problem is the registration of consecutive range images provided by a laser range finder. This paper proposes a method that uses

normal vectors as features to guide the matching phase of the ICP. Results show that the proposed method is reliable and finds better results than conventional ICP.

References

1. Besl, P.J., Mckay, N.D.: A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
2. Calderón, F., Ruiz, U., Rivera, M.: Surface-normal estimation with neighborhood reorganization for 3d reconstruction. In: CIARP, pp. 321–330 (2007)
3. Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. Image Vision Comput. 10(3), 145–155 (1992)
4. Chua, C.S., Jarvis, R.: 3D free-form surface registration and object recognition. International Journal of computer vision (1996)
5. Diosi, A., Kleeman, L.: Fast laser scan matching in polar coordinates. Journal of Robotics Research, 1125–1153 (2007)
6. Dudek, G., Jenkin, M.: Computational principles of mobile robotics. Cambridge University Press, New York (2000)
7. Freidman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Softw. 3(3), 209–226 (1977)
8. Lu, F., Milios, E.: Robot pose estimation in unknown environments by matching 2d range scans. Journal of Intelligent and Robotic Systems 18, 935–938 (1997)
9. Minguez, J., Montano, L., Lamiraux, F.: Metric–based iterative closest point scan matching for sensor displacement estimation. IEEE Transactions on Robotics 22(5), 1047–1054 (2006)
10. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, Heidelberg (1999)
11. Pulli, K.: Multiview registration for large data sets. In: Third International Conference on 3D Digital Imaging and Modeling (3DIM), p. 0160 (1999)
12. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: Third International Conference on 3D Digital Imaging and Modeling (3DIM) (June 2001)
13. Sharp, G., Lee, S., Wehe, D.: Icp registration using invariant features. IEEE Trans. on PAMI 24(1), 90–102 (2002)
14. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. Int. J. Comput. Vision 13(2), 119–152 (1994)

Multi-robot Exploration and Mapping Using Self Biddings and Stop Signals*

Juan C. Elizondo-Leal, Gabriel Ramírez-Torres,
and Gregorio Toscano Pulido

Centro de Investigación y Estudios Avanzados del IPN
Laboratorio de Tecnologías de Información
Carretera a Monterrey KM 6, Ciudad Victoria, Tamaulipas México
{jelizondo, grtorres, gtoscano}@tamps.cinvestav.mx

Abstract. Exploration and mapping of the environment is a critical issue in mobile robotics. In this paper we present an approach to multi-robot exploration and mapping, which is based in a distributed bidding process, where the bids are calculated by each robot. When a robot reaches the target it makes a decision by itself, which involves every one of the team members without lost of time in a totally distributed behavior and without the necessity of a central module. Furthermore, if the selected goal is discovered for another robot, this robot send a stop signal and the process of bidding is restarted.

The bid is computed such each robot will be assigned to a near unexplored frontier, while keeping the robots far from each other. The bid includes the distance of robot towards one of the frontiers, and the distances to the others robots toward their objectives plus the distance of these objectives toward the frontier. The numerical results demonstrate that our approach is efficient and fault tolerant.

1 Introduction

Exploration and mapping of the environment is a critical issue in mobile robotics. Since autonomous robots must be able to navigate effectively in environments, must possess the ability to build representations of those environments. In general, this requires a good exploration strategy. There exist several applications in which the exploration task is an integral part of the robotic mission, such as search and rescue, hazardous material handling, planetary exploration, etc. There are many advantages in mapping and exploration with multiple robots compared to a single robot, like fault tolerant, concurrency, which can greatly reduce the time needed for the exploration and merging of overlapping maps so the error can be diminished.

The problem of multi-robot exploration can be stated as: *n robots set out in unknown area equipped with sensing, localization and communication capabilities; design an efficient coordinated exploration algorithm to make a map of the unknown area.*

* This research was partially funded by project number 51623 from "Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas".

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 615–625, 2008.

Recently, bidding based approaches have arisen, which have demonstrated to reduce the time and traveling distance in previous works on multi-robot exploration [1] [2] [3] [4]. These approaches treat all robots as economic agents, which use a bidding mechanism in order to locally minimize the time and traveling distance. However, most of the proposals that are based on biddings require a central module that sometimes do not include all the team members.

In our approach, the bidding mechanism involves every one of the team members without lost of time in a totally distributed behavior and without the necessity of a central module. When a robot reaches the target it makes a decision by itself. Furthermore, if the selected goal is discovered by another robot, this robot send a stop signal and the process of bidding is restarted. The bid is computed such each robot will be assigned to a near unexplored frontier, while keeping the robots far from each other. The bid includes the distance of robot towards one of the frontiers, and the distances to the others robots toward their objectives plus the distance of these objectives toward the frontier. We uses occupancy grid maps to represent the environment. Each cell of such an occupancy grid map has one of three status: occupied, free or unknown. The numerical results demonstrate that our approach is efficient and fault tolerant.

The remainder of the paper is arranged as follows: Section 2 discusses previous work in the area of multi-robot exploration. Section 3 describe our approach to multi-robot exploration and section 4 presents numerical results that show the performance of our method. In section 5, we present our conclusions and discuss future research.

2 Related Work

There has been a wide variety of approaches to robotic exploration. Many approaches have been proposed for exploring unknown environments with single robots [5] [6] [7], much work remains to be done in coordinated multi-robot mapping and exploration. Yamauchi [8] in 1997 developed an approach for exploration with a single robot based on the concept of frontiers, regions on the boundary between open space and unexplored space. The basic idea is to keep the robot moving to the closest frontier cell. In [9] Yamauchi applies the concept of frontiers to a distributed, asynchronous multirobot exploration algorithm, where robots exchange perceptual information continuously. This approach brings the fault tolerance capability, however, there is no explicit coordination so that multiple robots may end up covering the same area and may even physically interfere with each other. Simmons et al. [1] in 2000 developed a multi-robot approach which uses a concept of frontier and a simple bidding protocol. The team requires a central agent to evaluate bidding from all the other robots to obtain the most information gain while reducing the cost or the travelling distance. Zlot et al. [2] in 2002 presented a multi-robot exploration approach which exploits a market architecture. The team of robots continuously negotiate with each other, improving their current plans and sharing information about which regions have and have not already been covered. Their algorithm can improve the reliability, robustness and efficiency of the exploration. However, the goal generation

algorithm they used is not as efficient as the frontier based and the negotiation process is complex. Baglietto et al. [10] in 2002 apply the concepts of entropy and frontier, where entropy quantifies the information gain associated with the frontier cell with the greatest information gains. Burgar et al. [3] in 2005 developed a coordinated multi-robot exploration algorithm which simultaneously considers the utility of unexplored areas and the cost for reaching these areas, using evidence grids maps to represent the environment and the concept of frontiers. They investigate the impact of the limited-range communication. They use a central module dynamically selected which receives the local maps and combine them into a single global map which then is broadcasted to all robots. But using a central module suffers of a single point fault. Sheng et al. [4] in 2006 presented a coordinated multi-robot exploration algorithm based on a distributed bidding model. They address the problem caused by the limited-range communication. The authors introduce a nearness measure, and therefore the robots tend to stay close to each other. This approach does not guarantee that all the members of the team participate in the bidding. Stachniss et al. [11] in 2006 developed a multi-robot exploration strategy which considers the semantic place information, they apply a classifier learned with AdaBoost. This approach required learning and consider similar places. Poernomo and Shell Ying [12] in 2006 proposed the separation of robots in the frontier selection, which reduce the chances that the robots interfere with each other. However, they assumed that the size of robot is small enough, such that the collision is negligible.

In general, the proposals that are based on biddings require a central module that sometimes do not include all the team members. So we propose an algorithm where the team members are all included without the necessity of selecting a central module.

3 Approach

Our approach uses a bidding mechanism based on target assignment to coordinate the actions of the robots, which is totally distributed since each robot calculates the bid of the others and its own bid, and make its own decision, furthermore, if the selected goal is discovered by another robot, this robot send a stop signal and the process of bidding is restarted. Thus adding efficiency and robustness, since it eliminates times of delay, single points of fault and reduces unnecessary travels.

During the exploration the sensing and mapping is made continuously by each robot and the bidding is carry on when a frontier is reached. The frontier is considered reached when the robot is at a short distance and the frontier is detected explored.

3.1 Bid Compute

The bid is computed such each robot will be assigned to a near unexplored frontier, while keeping the robots far from each other. The bid of all robots

(bid_k) includes the distance of robot towards its goal ρ_k plus the distance of this goal toward the frontier α_k, and the distances to the others robots toward their objectives (ρ_i) plus the distance of these objectives toward the frontier (α_i).

$$bid_k = \rho_k + \alpha_k - \sum_{i=0, i\neq k}^{n} (\rho_i + \alpha_i);\ \text{for } k = 0, \ldots, n-1 \tag{1}$$

where i and k represent the index of each one of the robots.

With the inclusion of the team data in the bid we managed to separate them from each other.

The current robot computes its bid and all the team's bids to reach each one of the frontiers. The complexity of the calculation of all the bids is $O((n^2-n)F)$, where n is the number of robots and F is the number of frontiers. Once all the bids have been computed the frontiers are sorted (according to its bid value associated to the present robot) for the process of bidding.

3.2 The Totally Distributed Bidding

This computation is used as follows: When a robot has reached its goal, it requests maps and targets of other robots and it waits their answers. Then the robot merges the maps to its local map and compute the bids to the nearest

Algorithm 1: Robot exploration

```
start_MappingOnline() My_Target = My_Pose Send_MyTarget while the
Map is not Complete
   broadcast(Request MapTarget)
   update_MyMap
   update_OtherRobotTarget
   detectFrontier()
   bid_compute()
   for i=0 to number of frontiers
      if I win this frontier
         My_Target=thisFrontier
         break
   if does not win any frontier
      wait other new pose
      continue
   send my target
   while not reach my target or not receive stop signal
      travel to My_Target
      if i view another target
         send stop signal
broadcast(MyMap)
```

Fig. 1. Algorithm for robot exploration

Algorithm 2: Robot communication

```
while the Map is not Complete
   idRobot=receiveData()
   if request map and target
      sendTarget(idRobot, My_Target)
      sendMap(idRobot, My_map)
   else if Receive target
      saveTarget(idRobot, Target)
      if(Target is equal My_Target)
         stopTravel
   else if Receive stop signal
      stopTravel
```

Fig. 2. Algorithm for robot communication

frontier. If the robot provide the best bid, then this robot wins the bidding and broadcast this frontier as its new goal. If this robot does not win, the process is repeated with the next nearness frontier until it wins, or no frontier is left. If the robot does not win any case then the robot stops until new data is received. Furthermore if the selected frontier is discovered by any robot, this robot send a stop signal and the process of bidding is restarted.

The pseudocode of our approach is shown in Figure 1 and run in parallel with the communication protocol in Figure 2. The complexity of this algorithm including calculation of bids is $O((n^2 - n)F^2)$, in the worst case where n is the number of robots and F is the number of frontiers.

Algorithm 3: Clustering frontier

```
PROCEDURE RecursiveDetection(x, y)
   for i=x-1 to x+1
      for j=y-1 to y+1
         if cell[i][j] is frontier and not considered before
            RecursiveDetection(i, j)
MAIN
   for i=0 to width of map
      for j=0 to height of map
         if cell[i][j] is frontier and not considered before
            RecursiveDetection(i, j)
            if size of cluster > threshold
               search geometrical center
               save frontier
```

Fig. 3. Algorithm for clustering frontier

3.3 Maps Updating

When a robot receives a map, it makes an average with its own map. The map computed by a robot can contain incorrect data, because of the own errors of the sensors. We exploit the advantage that multiple robots may produce more accurate maps by merging the overlapping information. When a robot receives a map, each point is overlapped with the local map, but if the local point is marked as explored, the point is averaged with the new point, so the errors can be diminished.

Once the map was updated, the detection of frontiers is carried out, which closer frontiers are considered as a single one located in the geometrical center of that cluster. The algorithm of clustering is showing in Figure 3.

4 Numerical Results

In this section we evaluate the performance of our approach. The simulator used in our approach is the player/stage simulator[1] [13] which models the interferences betwen robots. The environment is represented with a grid-based map. To carry out our experiments we simulated laser sensors with 360 degree of view in order to determine the occupied area and a ring of ultrasound sensors which determines the free area, the sensors radius range is 4 meters, mounted on the robot Pioneer 2-DX.

An example environment is shown in the figure 9. A number of robots are simulated by identical programs with each representing one robot. We performed four experiments.

First, we show the fault tolerance of this approach, using the map shown in figure 4. We identify two cases: the robot fails while traveling to its goal, and the robot fails during its bidding process. In the figure 5 the robot 2 has a failure during its travel, when the others robots begin the bid process they can not receive any new information from the robot 2 and they do not include it in its bidding process and can complete the task. In the figure 6 the robot 4 has some failure during its bid process, but this do not affect the other robots since each robot makes a decision by itself.

The results of the previous cases prove that the multi-robot exploration using self biddings and stop signals is tolerant to faults. When some robot has a failure, the other robots can continue to explore the unknown environment.

Second, we show the behaviour when a robot discovers another target. In the figure 7 the robot 4 discovers a goal assigned to robot 1, this last one is informed so it stops and bids for a new goal. The result of the second experiment prove that when stop signals are added the redundant work is reduced.

Third, we show the behaviour when a robot is added a long time after the beginning of exploration. In the figure 8 the robot 3 is added after 3.5 minutes, so when the other robots begin the bid process receives new information from

[1] Player is freely available under the GNU General Public License from http://playerstage.sourceforge.net.

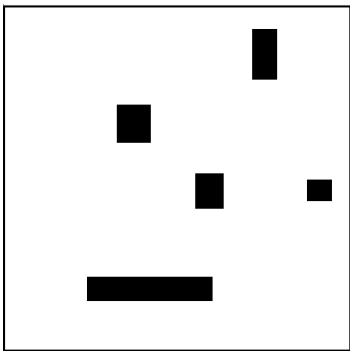
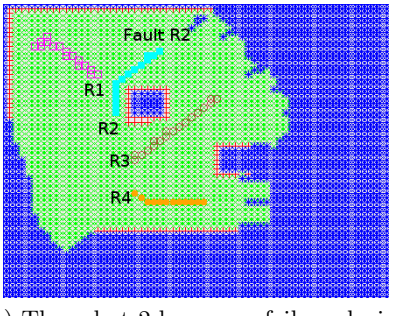
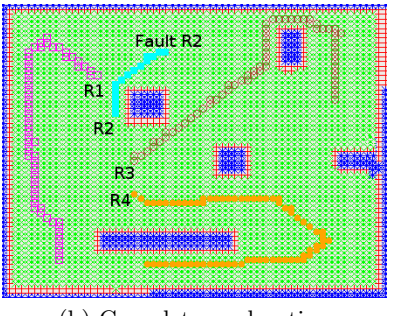


Fig. 4. Map used for the demonstration of fault tolerant (15 × 15 meters)

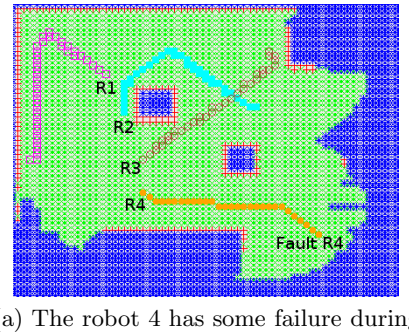


(a) The robot 2 has some failure during its travel

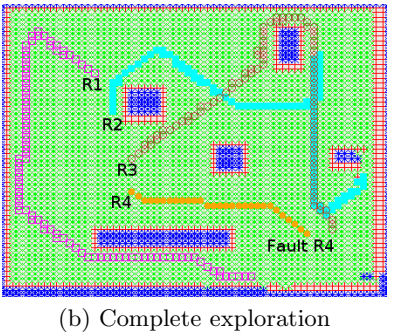


(b) Complete exploration

Fig. 5. Fault during travel

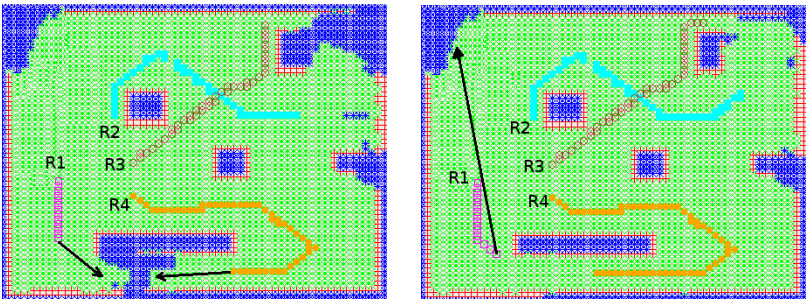


(a) The robot 4 has some failure during its bid process



(b) Complete exploration

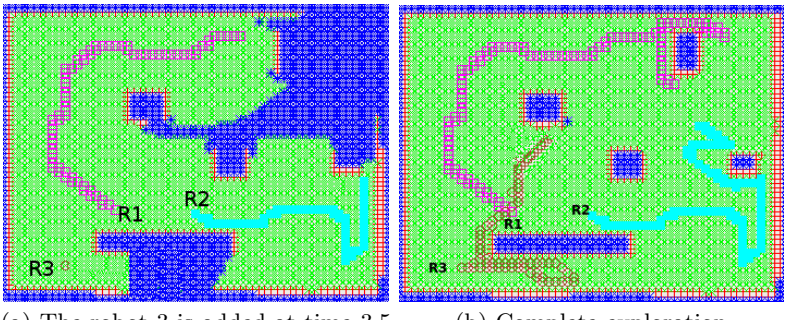
Fig. 6. Fault during bid process



(a) The robot 4 discovers the goal assigned to robot 1 and send stop signal

(b) The robot 1 selects another goal

Fig. 7. Discovery of the goal of another robot



(a) The robot 3 is added at time 3.5

(b) Complete exploration

Fig. 8. The robot 3 is added a time after the beginning of exploration

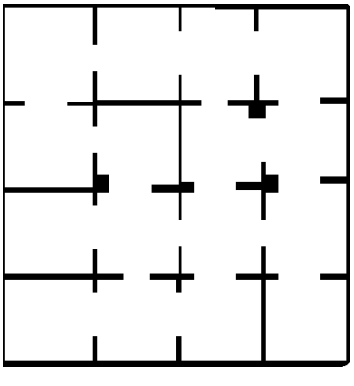
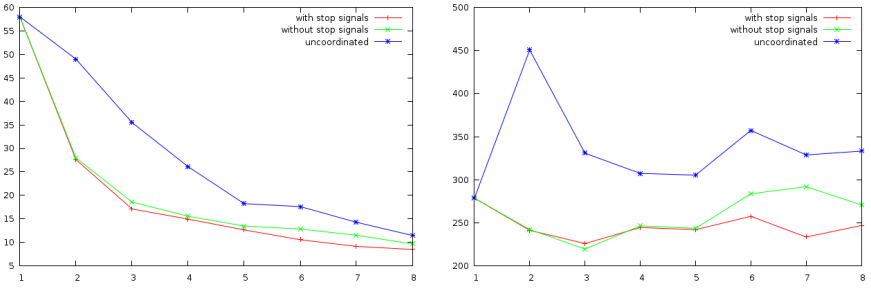
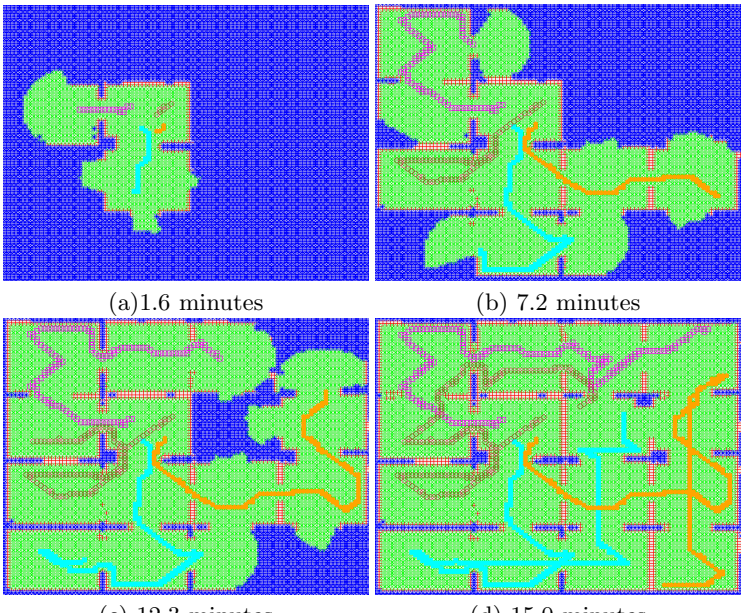


Fig. 9. Map used for the simulation (25 × 25 meters)



(a) Exploration time (b) Total distances

Fig. 10. Performances of the different coordination strategies for the environments shown in Figure 9



(a)1.6 minutes (b) 7.2 minutes

(c) 12.3 minutes (d) 15.0 minutes

Fig. 11. Simulated exploration with four robots

the robot 3 and they include it in its bidding process and may complete the task in a smaller time.

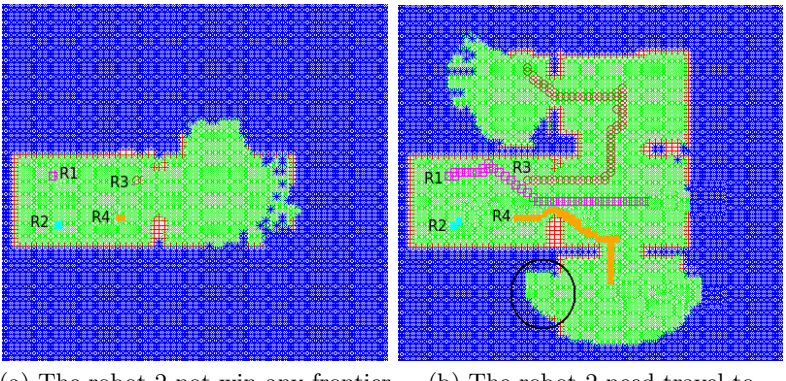
Fourth, we compared our approach with the technique where the process of bidding is eliminated and each robots goes to the nearest frontier, like used by Yamauchi. From now on, this approach will be denoted as uncoordinated exploration since it lacks a component that arbitrates between the robots whenever they choose the same frontier cells. Moreover we also compare it with our own

approach without stop signals. For each case and a fixed number of robots, ten test runs are conducted and the total exploration times are recorded. Figure 10 compares the total exploration time needed versus the number of robots and the total distances needed versus the number of robots. For each comparison of the strategies, the robot team was started at the same randomly chosen location. The map used in our experiments are shown in the figure 9. Screen shots of a simulation with four robots exploring the environment are shown in Figure 11 the robots started in the center of map.

The results of the fourth experiment demonstrate a great efficiency in the selection of the frontier to explore by every robot.

5 Future Work and Conclusions

This work presents a novel completely distributed multi-robot coordination algorithm to explore and map terrain, based on a market bidding process. The key idea of this approach is take into account the cost of reaching a unexplored frontier and the distances to the others robots and theirs objectives. The result is an efficient and fault tolerant approach. From the theoretical analysis and simulation experiments, it is shown that the method is feasible and can effectively distribute multiple frontiers to multiple robots in a totally uncentralized form.



(a) The robot 2 not win any frontier (time 0.5)

(b) The robot 2 need travel to a new frontier (time 3.5)

Fig. 12. The robot 2 stops some time while new frontier has discovered

Since sometimes the number of frontiers is less than the number of robots, some robots can remain stopped as shown in figure 12. This may cause over time in some robots. We believe that the process of exploration can be improved by adding an opportunistic distribution, so these robots follow at a safe distance other robots, such if the leader robot discover a new frontier, the follower will be ready to take it. At present, we are working in its development of such strategy.

References

1. Simmons, R.G., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H.: Coordination for multi-robot exploration and mapping. In: AAAI/IAAI, pp. 852–858 (2000)
2. Zlot, R., Stentz, A., Dias, M., Thayer, S.: Multi-robot exploration controlled by a market economy. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3016–3023 (2002)
3. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. IEEE Transactions on Robotics and Automation 21, 376–386 (2005)
4. Sheng, W., Yang, Q., Tan, J., Xi, N.: Distributed multi-robot coordination in area exploration. Robot. Auton. Syst. 54, 945–955 (2006)
5. Dudek, G., Jenkin, M., Milios, E., Wilkes, D.: Robotic exploration as graph construction. IEEE Transactions on Robotics and Automation, 859–865 (1991)
6. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence 99, 21–71 (1998)
7. Albers, S., Kursawe, K., Schuierer, S.: Exploring unknown environments with obstacles. In: SODA 1999: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms, Philadelphia, PA, USA, pp. 842–843. Society for Industrial and Applied Mathematics (1999)
8. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: CIRA 1997: Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Washington, DC, USA, p. 146. IEEE Computer Society, Los Alamitos (1997)
9. Yamauchi, B.: Decentralized coordination for multirobot exploration. Robotics and Autonomous Systems 29, 111–118 (1999)
10. Baglietto, M., Paolucci, M.S., Zoppoli, R.L.: Information-based multi-agent exploration. In: Proceedings of the Third International Workshop on Robot Motion and Control, RoMoCo 2002, November 9-11, 2002, pp. 173–179 (2002)
11. Stachniss, C., Martínez-Mozos, O., Burgard, W.: Speeding-up multi-robot exploration by considering semantic place information. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, pp. 1692–1697 (2006)
12. Poernomo, A., Ying, H.S.: New cost function for multi-robot exploration. In: 9th International Conference on Control, Automation, Robotics and Vision, ICARCV 2006, December 5-8, 2006, pp. 1–6 (2006)
13. Gerkey, B.P., Vaughan, R.T., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: Proc. of the Intl. Conf. on Advanced Robotics (ICAR), July 2003, pp. 317–323 (2003)

Rhythmic Locomotion Control of Humanoid Robot

Carlos Antonio Acosta Calderon, Rajesh Elara Mohan, and Changjiu Zhou

Advanced Robotics and Intelligent Control Centre,
School of Electrical and Electronic Engineering,
Singapore Polytechnic, 139651, Singapore
CarlosAcosta@sp.edu.sg

Abstract. This paper presents a structure of a neural oscillator which is used as Central Pattern Generator for a biped robot locomotion. This architecture is suitable to generate different rhythmic motions, according to the amplitude and frequency of the system. The presented approach uses two reflexes to attain the stability in the sagittal and coronal planes. Results of the system on a simulated biped robot and a real robot show the feasibility of this approach.

1 Introduction

Many biped locomotion algorithms use Zero Moment Point (ZMP) as criterion for off-line motion planning. These ZMP-based methods require a well-defined model of the dynamics and a prior knowledge of the environment. Our previous work in robotic biped locomotion includes approaches such as kinematics, dynamics, fuzzy logic, neural networks, genetic algorithms, and optimization methods [1,2,3,4]. Most of these approaches are for off-line locomotion based mainly on the ZMP. This paper studies on-line locomotion based on a biological inspired paradigm call Central Pattern Generators (CPG).

Neurobiologists pointed out clear evidence of CPGs in the animal's brain, which produce periodic movements in their locomotion. Therefore, GPGs encode rhythmic trajectories as limit cycles of nonlinear dynamical systems, typically systems of coupled nonlinear oscillators. CPG offers multiple interesting features such as the stability properties of the limit cycle behaviour (i.e. perturbations are quickly forgotten), the smooth on-line modulation of trajectories through changes in the parameters of the dynamical system, and entrainment phenomena when the CPG is coupled with a mechanical system. It is not prove that humans use a simple CPG based locomotion as the lamprey, but human system is comparable to a CPG for locomotion.

The main advantage of the CPG-based methods, compared to those using the ZMP, is that, there is not need of a perfect knowledge of the robot's dynamics. CPG-based methods are more general and adaptive for robotic applications involving rhythmic motion with interaction between the system and the environment. Examples of these applications include biped locomotion [5,6,7], quadruped locomotion [8,9], juggling [10], and drumming [11].

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 626–635, 2008.

Neural oscillators based on CPG are using the concept of limit cycles, which are very convenient for locomotion since they can return to their stable state after a small perturbation and are almost not influence by the change in the initial conditions. However, it is difficult to design interconnection and feedback pathways of neural oscillators, and to manually tune all open parameters in order to achieve the desired behaviour.

This paper presents a system that employs a CPG for the online generation of periodic trajectories to the control of biped locomotion in a simulated robot. The rest of this paper is organized as follows. Section 2 presents the humanoid robot and the simulator used for the experiments. Section 3 introduces the structure of the gait generator for the humanoid robot. In Section 4 the experimental results are presented. Finally, Section 5 summarizes the paper.

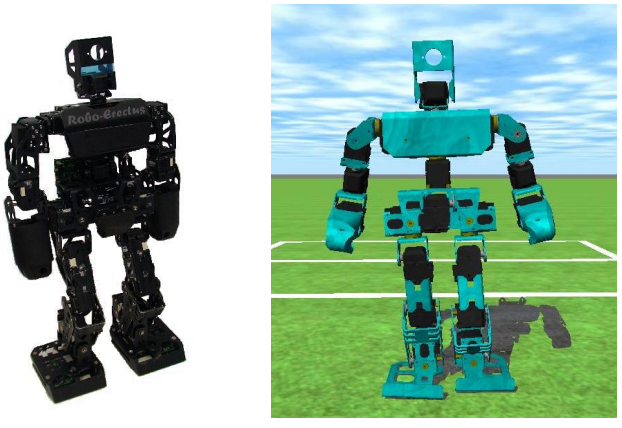
2 Humanoid Robot

The Humanoid Robot employed for the experiments is the Robo-Erectus Jr-AX. The Robo-Erectus project is developed in the *Advanced Robotics and Intelligent Control Centre (ARICC)* of Singapore Polytechnic. Robo-Erectus Jr-AX is a humanoid robot used to study tasks that required human like motions [12]. One of these applications is RoboCup, robots playing soccer. Robo-Erectus is one of the foremost leading soccer-playing humanoid robots in the RoboCup Humanoid League in KidSize. For more detailed information about the Robo-Erectus humanoid soccer robots, please refer to the team's website www.robo-erectus.org. The development of Robo-Erectus has gone through many stages either in the design of its mechanical structure, electronic control system and gait movement control. Robo-Erectus is able to perceive different colours and to track them. The robot also contains a dedicated processor used to control its behaviour, wireless communication with the control PC and teammates, and a sub-system to control sensors and actuators.

Figure 1 shows the design of the humanoid robot Robo-Erectus. Its human-like body has a height of 50cm and weight of just 3.2kg, including batteries. Robo-Erectus has a total of 24 degrees of freedom (See Table 1). Each degree of freedom uses as actuator a *Dinamixel DX-117 Digital Servomotor.* These servomotors have a typical torque of $28.89kg \cdot cm$ and a speed of $0.172sec/60°$. Each smart actuator has a micro-controller in charge of receiving commands and monitoring the performance of the actual motor. An RS485 serial network connects all the servomotors to a host processor which, sends positions and receives the current data (angular positions, speed, voltage, and temperature) of each actuator.

The robot is equipped with independent processors for image processing, behaviour control and sensor/motor processing. Communication from a PC or other robots can be achieved by WIFI or RS232.

Robo-Erectus has five main sensors: a *camera* to capture images, a *tilt* sensor to recognize whether it is standing or falling down, a *compass* to detect its own



(a) (b)

Fig. 1. The humanoid robot Robo-Erectus Junior-AX and its simulator Virtual-RE

Table 1. Physical Specifications of the degrees-of-freedom for the REJr-AX

Joint	Roll	Pitch	Yaw	
Head		✓	✓	
Body		✓	✓	
Shoulder	✓	✓		
Upper Arm			✓	
Elbow		✓		
Hip	✓	✓	✓	
Knee		✓		
Ankle	✓	✓		

orientation, a couple of *ultra–sonic* sensors to measure distance from front and rear objects, and finally a pair of gyroscopes to measure the rate of the motion.

The simulator for Robo-Erectus Jr-AX is called *Virtual-RE* [13]. Virtual-RE provides several possibilities of visualization and interaction with a simulated environment. The simulation of rigid body dynamics is done through *Open Dynamics Engine (ODE)*, which has a wide variety of features and has been used successfully in many other projects [14]. The visualization as well as the computation of camera image is based on OpenGL.

The simulator is a client/server-based system, which offers the possibility of halting or stepwise executing the whole simulation without any concurrencies. It allows also a more comprehensive debugging of the executed robot software. The simulation kernel models the robots and the environment, simulates sensor readings, and executes commands given by the controller or the user. The user interface is responsible for the display of information and for the interaction with the user. The controller implements the behaviour control. In each simulation

step, the controller reads the available sensors, plans the next action, and sets the actuators to the desired states.

Virtual-RE provides each robot with a set of simulated sensors, i.e. tilt, compass, gyroscopes, camera images, and motor feedback. The camera sensor generates an RGB24/YUV420 image of 320x240 pixels. The motors state are also simulated as in the real robot with feedback information that includes the joint angles as well as the velocities of motors.

3 Structure of the Gait Generator

Many neural oscillator models have been proposed resembling the animal's brain neurons in the CPG, such as McCulloch-Pitts, Leaky integrator, Rayleigh, Matsuoka, and Hopf. A very interesting property of non-linear oscillators is the concept of *limit cycle*. A limit cycle is an isolated close trajectory of the system, which can be stable or unstable. Trajectories close to the limit cycle are attracted by it. Therefore, signals generated under this principle can return their stable position after small perturbations.

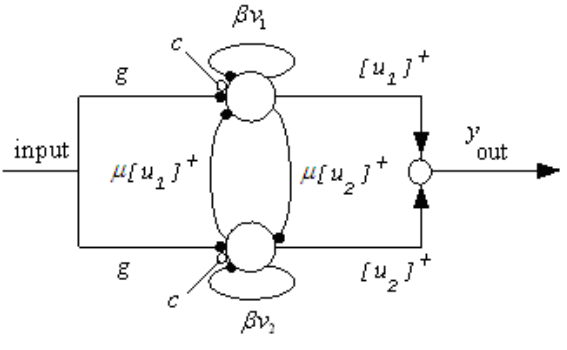


Fig. 2. Matsuoka oscillator consists of two neurons, which inhibited their output alternately. The black circles correspond to inhibitory connections and white circles to excitatory connections.

The Matsuoka oscillator adapts to the natural dynamics of the controlled object and hence provides a means of energy-efficient actuation. This model is based on the mutual inhibition of two neurons that generate a periodic signal as output. The self-inhibition is governed by the βv_i connections while mutual inhibition is done through the $\mu[x_i]^+$ connections. The Matsuoka model is shown in Fig. 2 and its behaviour is described by the following equations. The output of this model is presented in Fig. 3.

$$t_1\dot{u}_1 = c - u_1 - \beta v_1 - \mu[u_2]^+ - \sum h_j[g_j]^+ \tag{1}$$

$$t_2\dot{v}_1 = [u_1]^+ - v_1 \tag{2}$$

$$t_1\dot{u}_2 = c - u_2 - \beta v_2 - \mu[u_1]^+ - \sum h_j[g_j]^- \tag{3}$$

$$t_2\dot{v}_2 = [u_2]^+ - v_2 \tag{4}$$
$$[u_k]^+ = max(0, u_k) \tag{5}$$
$$Y_{out} = [u_1]^+ - [u_2]^+ \tag{6}$$

Each neuron is represented by two equations, the associated variable u correspond to the firing rate of the neuron, while the variable v represents the neuron's self-inhibition. The neurons excite and inhibit alternatively producing the oscillation described by Y_{out} in (6). The parameter c is directly proportional to the output and allows controlling the amplitude of the output signal. The coefficient β indicates the effect of the adaptation; and μ represents the strength of inhibition connection between neurons. The terms $\sum h_j[g_j]^+$ and $\sum h_j[g_j]^-$ define the external input in the system. The frequency of the oscillator when no external input is applied is determined by the time constants t_1 and t_2.

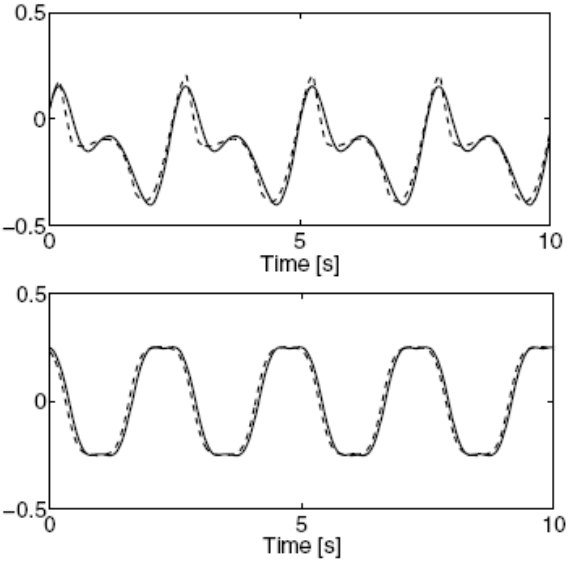


Fig. 3. Oscillation generated by the Matsuoka model in the absence of external input. Both signals are for hip control.

Another important aspect of the CPG is that it allows easy modulation of the amplitude and the frequency of the trajectory. Because of the properties of coupled oscillators, modulation of these parameters is always smooth and thus interesting for trajectory generation in a robot.

The CPG system was developed for the Robo-Erectus Jr-AX described in Section 2. Since the aim is gait generation only 11 joints were selected for locomotion: 1 for trunk, 2 for each hip, 1 for each knee, and 2 for each ankle (see Fig. 4). For each DOF an oscillator is implemented, the oscillators are coupled by adding coupling terms to the Equations 1 and 3, and a phase difference is also introduced between the oscillators.

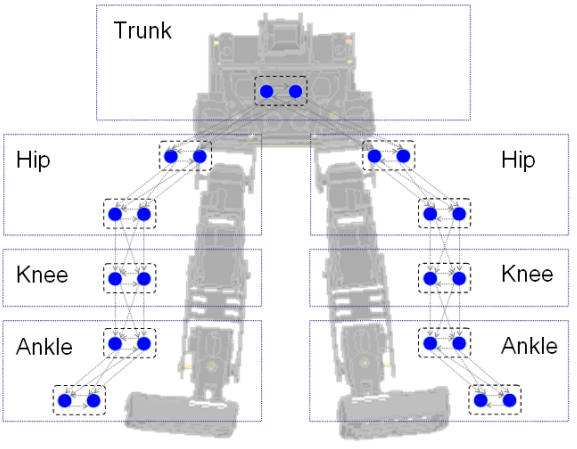


Fig. 4. Couplings between the joints for REJr-AX based on Matsuoka oscillators

The CPG can thus produce trajectories that are stable to perturbations. This can be useful when integrating sensory feedback in the CPG to be sure that the sensory information will be forgotten as soon as it disappears from the environment. Sensor feedback to a CPG is call reflex, which is used to handle perturbations and balance control. The model presented here is close loop since it integrates the feedback from the reflex system to the CPG. The purpose of the feedback is to attain the balance of the robot during locomotion.

3.1 Sagittal Stability

In humans the stability is achieved by the vestibular system, which calculate the tilt of the body and triggers the activation of muscles to keep balance. In similar way, the two gyroscopes located in the trunk of the robot's body calculate the sagittal and coronal tilt of the body. When this tilt is increasing the robot compensate the motion in the opposite direction.

The feedback is added to the system to guarantee the stability of the tilt of the robot in the sagittal plane. To achieve this, the amplitude of the signal must be modified in some DOFs to compensate the variation of the tilt angle. The feedback is added to the oscillators representing hip2 and knee joints. Equations (7) to (12) describe the oscillator once the sagittal feedback is added. The coefficient k is the gain for the sagittal tilt and it is null for each oscillator except hip2 and knee.

3.2 Coronal Stability

One important feature is the lateral/coronal stability during the human gait. Lateral stability will produce a human-like locomotion as well as efficient use of the energy in the gait. The robot's gyroscopes calculate the lateral tilt of

the body, when this tilt is increasing the robot compensate the motion in the opposite lateral direction.

Two main joints are controlling the lateral direction in the robot's locomotion: hip1 and ankle2. These joints have a direct influence in the roll angle (see Fig. 4). Preserving the ankle parallel to the ground is essential to attain a correct orientation when touching the ground. Hence the ankle should present an opposite effect from the hip. Figure 3 shows the symmetry in the switch of trajectories for the hips and ankles.

The effects of the reflexes for sagittal and coronal tilt are projected to the oscillator model as described below. This ensures that the phase of the oscillation is preserved since the amplitude of the trajectory is relevant for this study.

$$\begin{aligned} t_1\dot{u}_{1,i} &= c_i - u_{1,i} - \beta v_{1,i} - \mu[u_{2,i}]^+ \\ &\quad -\alpha_{i,i+1}\gamma_i[u_{1,i+1}]^+ - \alpha_{i,i+1}(1-\gamma_i)[u_{2,i+1}]^+ \\ &\quad +k_i\theta + \lambda_i\psi \qquad (7) \\ t_2\dot{v}_{1,i} &= [u_{1,i}]^+ - v_{1,i} \qquad (8) \\ t_1\dot{u}_{2,i} &= c_i - u_{2,i} - \beta v_{2,i} - \mu[u_{1,i}]^+ \\ &\quad -\alpha_{i,i+1}\gamma_i[u_{2,i+1}]^+ - \alpha_{i,i+1}(1-\gamma_i)[u_{1,i+1}]^+ \\ &\quad +k_i\theta + \lambda_i\psi \qquad (9) \\ t_2\dot{v}_{2,i} &= [u_{2,i}]^+ - v_{2,i} \qquad (10) \\ [u_{k,i}]^+ &= max(0, u_{k,i}) \qquad (11) \\ Y_{out,i} &= [u_{1,i}]^+ - [u_{2,i}]^+ \qquad (12) \end{aligned}$$

The oscillator i is coupled in one direction with the oscillator $i+1$, thus each neuron of i is coupled with both neurons in $i+1$. The parameter $\alpha_{i,i+1}$ corresponds to the global coupling strength between oscillator i and $i+1$. The control value is relative to the coupling of each neuron [15].

4 Experimental Implementation

This section presents the experiments performed with the CPG controller presented in Fig. 4 and described by (7) to (12). The sensory feedback was integrated with the controller to achieved the sagittal and coronal stability described in Section 3. The lower level control is done by a PD controller. The generated control signal were used to control the Virtual-RE as well as REJr-AX (for details of the robot or its simulator see Section 2).

A phase difference of π was set between legs to achieve the symmetry between the joints of each leg. According to the amplitude of the signal from the different DOFs in Fig. 5, we can see that hip1 has a bigger impact than ankle2 in the coronal stability. In similar way hip2 contribution is more significant than that one from the knee. These joints, hip1 and hip2 are crucial to produce a stable locomotion.

The coupling parameters are crucial for the performance of the system. The reflex signals from the gyroscopes generated changes in the frequency of the

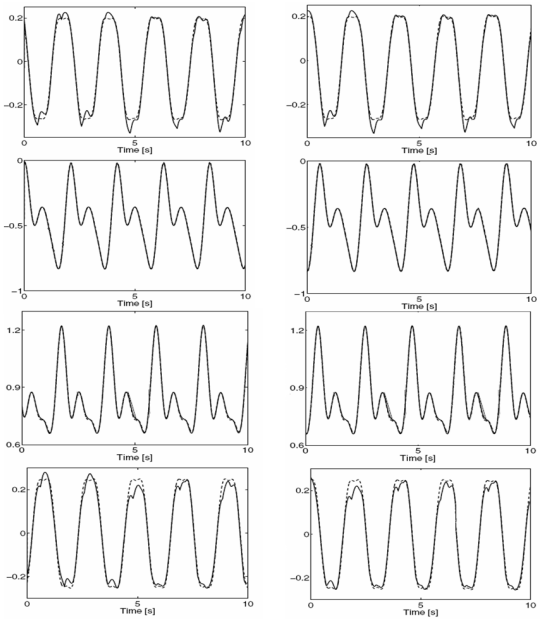


Fig. 5. Oscillation signals for the leg joints. The dotted line represents the original signal, and the solid line is the actual GPG output. Both legs are symmetrical only with a different phase. Joints from top to bottom: thigh, hip, knee, and ankle for right and left legs; for joint distribution see Fig. 4.

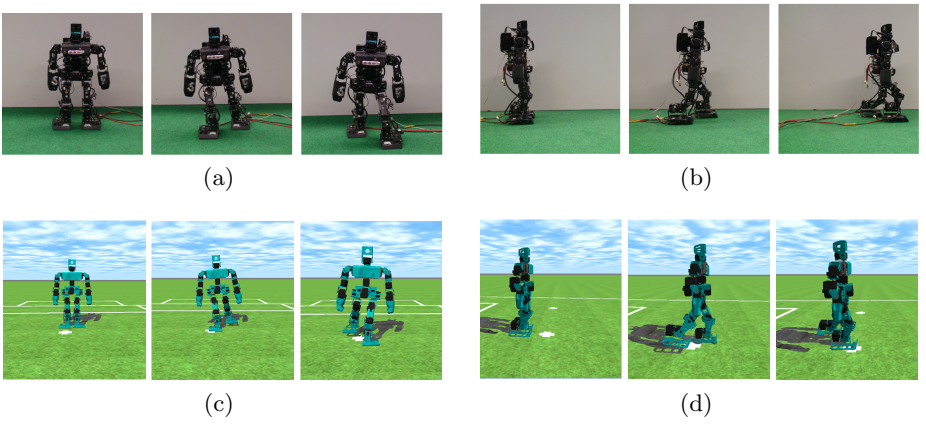


Fig. 6. Robo-Erectus walking with feedback for sagittal and coronal balance. Top figures the generated walking gait on the simulator. Bottom figures display the actual performance with Robo-Erectus Jr-AX.

oscillators that received the feedback, and these influence the rest of the system due to the coupling.

Several locomotion gaits could be obtained by tuning the phase difference of the oscillators in the CPG. Phase difference can be adjusted by modifying the weights of the couplings between the oscillators γ.

Preliminary studies were performed with the simulator of REJr-AX, and after the stability analysis tested with the real REJr-AX. Figure 6 shows the REJr-AX walking forward using the signals presented in Fig. 5. The variation of the amplitude on the control signal generates different gaits. However, a stability analysis reveals that not all the generated gaits could be used to drive either the simulated or the real robot. One reason for this is that motion produced causes the robot to swing faster than it can compensate using the gyroscope information. Changing the values for the compensation helps to reduce the ample of the signal and regain stability. Methods like GA could be applied to find optimal values.

External perturbations were applied to test the balance of the system. The robot presented a resistance against small perturbations. Unfortunately, greater forces make the robot fall, a great force produce a big change in the tilt value and the system cannot adapt to quickly to respond to a sudden drastic change.

5 Conclusions

Biped locomotion offers several challenges; one of them is to have a system that can produce on-line motion, while it is somehow robust to unknown environments. Central Pattern Generator is a biologically inspired method to generate periodic movements such as walking. Neural oscillators based on CPG can adapt and manage to small perturbations.

This paper presents a CPG system that uses two reflexes to attain the stability in the sagittal and coronal planes. The system was tested in the simulator Virtual-RE and the actual robot Robo-Erectus Jr-Ax. The results showed that the frequency can be modified to reach different locomotion speeds. It was also concluded that stability could be achieved by integrating a gyroscope signal to the CPG controller. However, different speeds require different coefficients for the compensation.

Future work involves the use of a force sensor to module the signal. It is important to study the switch between different speeds in the system. Future work should also involve the adaptation of the system to a different environment.

Acknowledgements

The authors would like to thank staff and students at the Advanced Robotics and Intelligent Control Centre (ARICC) and higher management of Singapore Polytechnic for their support in the development of the humanoid robots and the simulator and analysis tools.

References

1. Hu, L., Zhou, C., Sun, Z.: Biped gait optimization using spline function based probability model. In: Proc. of IEEE Int. Conf. Robotics and Automation (ICRA) (2006)
2. Hu, L., Zhou, C., Sun, Z.: Biped gait optimization using estimation of distribution algorithm. In: Proc. of IEEE-RAS Int. Conf. Humanoid Robots (2005)
3. Tang, Z., Zhou, C., Sun, Z.: Gait synthesizing for humanoid penalty kicking. In: Proc. of Int. Conf. Engineering Applications and Computational Algorithms (DCDIS), Ontario, Canada (2003)
4. Zhou, C., Meng, Q.: Dynamic balance of a biped robot using fuzzy reinforcement learning agents. Fuzzy Sets and Systems (2003)
5. Endo, G., Morimoto, J., Nakanishi, J., Cheng, G.: An empirical exploration of a neural oscillator for biped locomotion. In: Proc. of IEEE Int. Conf. Robotics and Automation (ICRA), pp. 3036–3042 (2004)
6. Hase, K., Yamazaki, N.: Computational evolution of human bipedal walking by a neuro-musculo-skeletal model. Artificial Life and Robotics 3, 133–138 (1999)
7. Miyakoshi, S., Taga, G., Kuniyoshi, Y., Nagakubo, A.: Three dimensional bipedal stepping motion using neural oscillators-towards humanoid motion in the real world. In: Proc. of IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS), pp. 84–89
8. Miyakoshi, S., Yamakita, M., Furuta, K.: Juggling control using neural oscillators. In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 1186–1193 (1994)
9. Fukuoka, Y., Kimura, H., Cohen, A.: Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. International Journal of Robotics Research 22, 187–202 (2003)
10. Kang, T.H., Koo Ig Mo, S.Y.K., Vo, G., Trong, T.D., Lee, C.M., Choi, H.R.: Control of quadruped walking robot based on biological inspired approach. In: Proc. of Advances in Climbing and Walking Robots (CLAWAR), pp. 242–251 (2007)
11. Kotosaka, S., Schaal, S.: Synchronized robot drumming by neural oscillator. In: Proc. of Int. Symposium on Adaptive Motion of Animals and Machines (2000)
12. Acosta Calderon, C.A., Zhou, C., Yue, P.K., Wong, M., Rajesh Elara, M.: A distributed embedded control architecture for humanoid soccer robots. In: Proc. of Advances in Climbing and Walking Robots (CLAWAR), Singapore, pp. 487–496 (2007)
13. Acosta Calderon, C.A., Mohan, R.E., Zhou, C.: A humanoid robotic simulator with application to robocup. In: Proc. of IEEE Latin America Robotic Simposium/Congreso Mexicano de Robotica (LARS), Mexico (2007)
14. Smith, R.: Open dynamics engine (2008), http://www.ode.org
15. Williamson, M.M.: Robot Arm Control exploiting Natural Dynamics. PhD thesis, MIT (1999)

Strong Probabilistic Planning

Silvio do Lago Pereira, Leliane Nunes de Barros, and Fábio Gagliardi Cozman

Institute of Mathematics and Statistics - University of São Paulo
Rua do Matão, 1010 - São Paulo, Brazil

Abstract. We consider the problem of synthesizing policies, in domains where actions have *probabilistic* effects, that are optimal in the *expected-case* among the optimal *worst-case strong* policies. Thus we combine features from nondeterministic and probabilistic planning in a single framework. We present an algorithm that combines dynamic programming and model checking techniques to find plans satisfying the problem requirements: the strong preimage computation from model checking is used to avoid actions that lead to cycles or dead ends, reducing the model to a Markov Decision Process where all possible policies are strong and worst-case optimal (*i.e.*, successful and minimum length with probability 1). We show that backward induction can then be used to select a policy in this reduced model. The resulting algorithm is presented in two versions (enumerative and symbolic); we show that the latter version allows planning with extended reachability goals.

1 Introduction

Optimal behavior is a very desirable property of autonomous agents and has received much attention in the automated planning community over the last years. Particularly, when agent's actions have probabilistic effects, Markov Decision Processes (MDPs) provide a simple and elegant framework for synthesizing optimal expected-case policies [Puterman 1994]. In this framework, the execution of each action produces a reward and the objective is to maximize the expected value of a sum of exponentially discounted rewards received by the agent. The classical MDP formulation has some important properties: it is subject to the principle of local optimality, according to which the optimal action for a state is independent of the actions chosen for other states, and optimal policies are stationary and deterministic. These properties translate into very efficient dynamic-programming algorithms [Bellman 1957], e.g., *value-iteration* [Puterman 1994]. However, as pointed in [Dolgov and Durfee 2005], there are some situations where the classical MDP formulation proves inadequate, because it can be very difficult (if not impossible) to fold all the relevant feedback from the planning environment into a single reward function.

In this paper, we use *strong probabilistic planning* to refer to situations where one wishes to produce a policy that guaranteedly reaches a goal state, with minimum number of steps in the worst-case and maximum reward in the expected-case. For instance, such situation can happen in a domain where the agent should deliver a mail, minimizing the time spent in the worst-case, while maximizing its expected reward. To get

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 636–652, 2008.

some intuition on the difficulty of combining these constraints in the classical MDP framework, consider the situations depicted in Figure 1, where the agent's goal is to deliver a mail in room s_2. For both situations, *value-iteration* synthesizes the optimal expected-case policy $\pi = \{(s_0, a)\}$. However, by following π in the first one (Figure 1-a), the agent cannot guarantee that the mail will be delivered (since the execution of action a in state s_0 can lead to the dead-end s_3); and, in the second situation (Figure 1-b), the agent cannot guarantee a minimum number of steps for delivering the mail (since the execution of action a in state s_0 can require an unbounded number of steps before reaching the goal state s_2). Thus, in the first case, the agent incurs the risk of failing to deliver the mail; while in the second case, the agent incurs the risk of delivering the mail when it is no more useful. As we show in this paper, strong probabilistic planning is a novel framework to implement *risk-averse* agents, by combining decision-theoretic concepts [Boutilier, Dean, and Hanks 1999] with model checking (MC) techniques [Muller-Olm, Schimidt, and Steffen 1999].

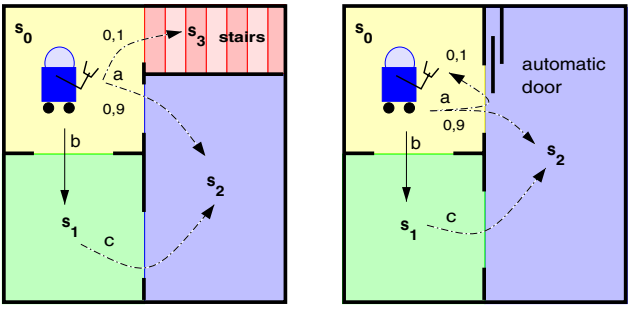


Fig. 1. A delivering agent domain

We should note that, at least in principle, we are dealing with constrained MDPs; that is, MDPs with additional constraints on functionals over trajectories [Puterman 1994, page 229]. However, there are two difficulties in directly resorting to the theory of constrained MDPs [Altman]. First, it is not entirely obvious how to model the constraint that the policy should be optimal in the worst-case. Second, general constrained MDPs fail the dynamic programming principle and, therefore, cannot be solved by backward induction [Altman]. On the other hand, as we show in this paper, strong probabilistic planning admits a backward reasoning scheme and, thus, yields better results than a general approach based on constrained MDPs.

The remainder of this paper is organized as follows. First, we present a review of MC/MDP-based planning, that contributes with a unifying perspective that seeks to compare its algorithms on a common ground. The insights produced by this comparison steer us to our algorithm for strong probabilistic planning. We describe two versions of this algorithm, one enumerative and one symbolic, and prove that they return a policy that is expected-case optimal among the strong policies which are worst-case optimal. Finally, we discuss our implementation and present our conclusions.

2 Approaches for Planning under Uncertainty

In this section we give an unified perspective on planning under uncertainty as we present the necessary background for the latter sections. We would like to stress that a combined presentation of nondeterministic and probabilistic planning is rarely found in the literature; thus, we devote considerable space to this issue. The approaches of interest here are:

- *Model Checking* (MC), used for planning under nondeterministic uncertainty models; and
- *Markov Decision Processes* (MDP), used for planning under probabilistic uncertainty models.

Both approaches are very attractive, but for different reasons: the main advantage of MC-based approach is the possibility of exploiting the knowledge about the structure of the planning domain (i.e., the state-transition graph induced by the actions); while the main advantage of MDP-based approach is the possibility of exploiting the knowledge about the probability of the actions' effects.

2.1 Nondeterministic Planning Based on MC

The basic idea underlying nondeterministic planning based on MC is to solve problems model-theoretically [Giunchiglia and Traverso 1999].

Definition 1. *A nondeterministic planning domain is a tuple* $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$*, where:*

- $\mathcal{S}$ *is a finite nonempty set of states;*
- $\mathcal{A}$ *is a finite nonempty set of actions;*
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \{0, 1\}$ *is a state transition function.* ◆

We assume that $\mathcal{A}$ contains a *trivial* action τ such that, for all $\sigma \in \mathcal{S}$, we have that $\mathcal{T}(\sigma, \tau, \sigma') = 1$ if and only if $\sigma = \sigma'$. So, when an agent executes action τ in a certain state, it always remains in this same state. Intuitively, action τ represents the fact that the agent can choose do nothing in any state. Given a state σ and an action α, the set of α-*successors* of σ, denoted by $\mathcal{T}(\sigma, \alpha)$, is the set $\{\sigma' : \mathcal{T}(\sigma, \alpha, \sigma') = 1\}$.

A *policy* in a nondeterministic planning domain $\mathcal{D}$ is a *partial* function $\pi : \mathcal{S} \mapsto \mathcal{A}$, that maps states to actions. A *nondeterministic policy* is a partial function $\pi : \mathcal{S} \mapsto 2^{\mathcal{A}} \setminus \varnothing$, that maps states to sets of actions. The set $\mathcal{S}_\pi$ of states reached by a policy π is $\{\sigma : (\sigma, \alpha) \in \pi\} \cup \{\sigma' : (\sigma, \alpha) \in \pi \text{ and } \sigma' \in \mathcal{T}(\sigma, \alpha)\}$. Given a policy π, the corresponding *execution structure* $\mathcal{D}_\pi$ is the subgraph of $\mathcal{D}$ that has $\mathcal{S}_\pi$ as set of states and that contains all transitions induced by the actions in policy π.

Definition 2. *A nondeterministic planning problem is a tuple* $\mathcal{P} = \langle \mathcal{D}, s_0, \mathcal{G} \rangle$*, where:*

- $\mathcal{D}$ *is a nondeterministic planning domain;*
- $s_0 \in \mathcal{S}$ *is the initial state;*
- $\mathcal{G} \subseteq \mathcal{S}$ *is a set of goal states.* ◆

Given a nondeterministic planning problem, we distinguish three kinds of solutions:

- a *weak solution* is a policy that may achieve the goal, but due to nondeterminism, is not guaranteed to do so. A policy π is a weak solution if some path in $\mathcal{D}_\pi$, starting from s_0, reaches a state in $\mathcal{G}$ [Cimatti et al. 1997].
- a *strong-cyclic solution* is a policy that always achieves the goal, under the fairness assumption that execution will eventually exit from all cycles. A policy π is a strong-cyclic solution if all paths in $\mathcal{D}_\pi$ starting from s_0 reach a state in $\mathcal{G}$ [Daniele, Traverso, and Vardi 1999].
- a *strong solution* is a policy that always achieves the goal, in spite of nondeterminism. A policy π is a strong solution if the subgraph $\mathcal{D}_\pi$ is acyclic and all paths starting from s_0 reach a state in $\mathcal{G}$ [Cimatti, Roveri, and Traverso 1998].

The strong nondeterministic planning algorithm. The strong nondeterministic planning algorithm, adapted from [Cimatti, Roveri, and Traverso 1998], allows us to synthesize policies that are guaranteed to reach a goal state, regardless of nondeterminism. This algorithm is correct, complete and returns an *optimal worst-case* strong policy π, in the sense that the longest path in $\mathcal{D}_\pi$ has minimal length.

```
StrongNondeterministicPlanning(P)
1 π ← ∅
2 π' ← {(σ, τ) : σ ∈ G}
3 while π ≠ π' do
4   S ← StatesCoveredBy(π')
5   if s_0 ∈ S then return π'
6   π ← π'
7   π' ← π' ∪ Prune(StrongPreimage(S), S)
8 return failure
```

The basic step in the nondeterministic planning algorithm is performed by the function StrongPreimage(S), which returns the set of pairs (σ, α) such that the execution of action α in state σ *necessarily* leads to a state inside S. This function is defined as following:

```
StrongPreimage(S)
1 return {(σ, α) : σ ∈ S, α ∈ A and ∅ ≠ T(σ, α) ⊆ S}
```

By iterating the strong preimage function from the set of goal states $\mathcal{G}$, the planning algorithm builds up a finite backward search tree (Figure 2). Because the set of states $\mathcal{S}$ is finite and this function is monotonic w.r.t. set inclusion, i.e., $\mathcal{G} \subseteq \text{StrongPreimage}^1(\mathcal{G}) \subseteq \text{StrongPreimage}^2(\mathcal{G}) \subseteq \cdots \subseteq \text{StrongPreimage}^n(\mathcal{G})$, after a finite number of iterations, a fixpoint is obtained. During this iterative process, the planning algorithm maps the states in the search tree to actions (or sets of actions) and, therefore, a policy is synthesized as a side effect. Furthermore, at each iteration, the set of states covered by the policy under construction π' is obtained by the following function:

```
StatesCoveredBy(π')
1 return {σ : (σ, α) ∈ π'}
```

Thus, if there exists a strong policy to reach a state in $\mathcal{G}$, from the initial state s_0, then in one of these iterations, the condition $s_0 \in \text{STATESCOVEREDBY}(\pi')$ is satisfied and the algorithm returns the policy π' as a solution to the planning problem. Finally, to avoid the assignment of new actions to states already covered in previous iterations (i.e., to avoid cycles and to guarantee optimal worst-case policies), the planning algorithm uses the following pruning function:

PRUNE(R, S)
1 `return` $\{(\sigma, \alpha) \in R : \sigma \in S\}$

As we can see, this function prunes from the set R (the strong preimage) every pair (σ', α) such that the state σ' is already in the set S (the covering of the current policy).

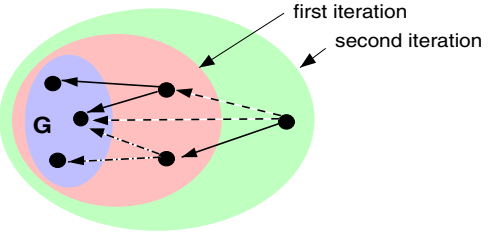


Fig. 2. A backward search tree built after two iterations of the strong preimage function

2.2 Probabilistic Planning Based on MDPs

The basic idea underlying probabilistic planning based on MDPs is to represent the planning problem as an optimization problem [Boutilier, Dean, and Hanks 1999].

Definition 3. *A probabilistic planning domain is a tuple $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$, where:*

- *$\mathcal{S}$ is a finite nonempty set of states;*
- *$\mathcal{A}$ is a finite nonempty set of actions;*
- *$\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ is a state transition function.* ◆

Given two states σ, σ' and an action α, the probability of reaching σ' by executing α in σ is $\mathcal{T}(\sigma, \alpha, \sigma')$. Furthermore, for each state $\sigma \in \mathcal{S}$, if there exists α and σ' such that $\mathcal{T}(\sigma, \alpha, \sigma') \neq 0$, then $\sum_{\sigma' \in \mathcal{S}} \mathcal{T}(\sigma, \alpha, \sigma') = 1$. Particularly, for the trivial action τ, we must have:

$$\mathcal{T}(\sigma, \tau, \sigma') = \begin{cases} 0 \text{ iff } \sigma \neq \sigma' \\ 1 \text{ iff } \sigma = \sigma' \end{cases}$$

Given a state σ, the set of *executable actions* in σ, denoted by $\mathcal{A}(\sigma)$, is the set $\{\alpha : \exists \sigma' \in \mathcal{S}$ such that $\mathcal{T}(\sigma, \alpha, \sigma') \neq 0\}$.

A *policy* in a probabilistic planning domain $\mathcal{D}$ is a *total* function $\pi : \mathcal{S} \mapsto \mathcal{A}$, that maps states to actions. Given a policy π, the corresponding *execution structure* $\mathcal{D}_\pi$ is the subgraph of $\mathcal{D}$ that has $\mathcal{S}$ as set of states and that contains all transitions induced by the actions in policy π.

Definition 4. *A probabilistic planning problem is a tuple* $\mathcal{P} = \langle \mathcal{D}, \mathcal{G} \rangle$*, where:*

- $\mathcal{D}$ *is a probabilistic planning domain;*
- $\mathcal{G} \subseteq \mathcal{S}$ *is a set of goal states.* ◆

A *reward function* $\mathcal{R} : \mathcal{S} \mapsto \mathbb{R}_+$ is a function that maps states to rewards. Intuitively, when the agent reaches a state σ it receives a reward $\mathcal{R}(\sigma)$. In the case of probabilistic planning for reachability goals, given a set of goal states $\mathcal{G}$, a Boolean reward function can be defined as following:

$$\mathcal{R}(\sigma) = \begin{cases} 0 \text{ iff } \sigma \in \mathcal{G} \\ 1 \text{ iff } \sigma \in \mathcal{G} \end{cases}$$

A reward is an "incentive" that attracts the agent to goal states. Moreover, to force the agent to prefer shortest paths to goal states, at each executed step, future rewards are discounted by a factor $0 < \gamma < 1$ (The use of such discount factor also guarantees convergence of fixpoint computations [Puterman 1994].). Hence, if the agent reaches a goal state by following a path with n steps, it receives a reward of γ^n. Since the agent wants to maximize its reward, it should minimize the expected length of paths to goal states.

The *optimal expected-value* of a state σ can be computed as the fixpoint of the following equation [Bellman 1957]:

$$v_n(\sigma) = \begin{cases} \mathcal{R}(\sigma) & \text{iff } n = 0 \\ \max\limits_{\alpha \in \mathcal{A}(\sigma)} \{\, g_n(\sigma, \alpha) \,\} & \text{iff } n > 0, \end{cases}$$

where the *expected gain* in state σ when action α is executed, denoted by $g(\sigma, \alpha)$, is defined as:

$$g_n(\sigma, \alpha) = \gamma \times \sum_{\sigma' \in \mathcal{S}} \mathcal{T}(\sigma, \alpha, \sigma') \times v_{n-1}(\sigma')$$

By selecting an action α that produces the optimal value for a state σ, for each $\sigma \in \mathcal{S}$, we can build an optimal policy:

$$\pi^\star(\sigma) = \arg \max_{\alpha \in \mathcal{A}(\sigma)} \{g_n(\sigma, \alpha)\}$$

A policy π is a *solution* for a probabilistic planning problem $\mathcal{P}$ if and only if π is an optimal policy for $\mathcal{P}$ [Ghallab, Nau, and Traverso 2004]. According to this definition, any probabilistic planning problem has a "solution", since it is always possible to find optimal policies for MDPs. Note, however, that this does not mean that such solution allows the agent to reach a goal state: *an optimal policy is independent of the initial state of the agent.*

The probabilistic planning algorithm. The probabilistic planning algorithm, based on the value-iteration method [Bellman 1957], allows us to synthesize *optimal expected-case* policies for probabilistic planning problems.

```
PROBABILISTICPLANNING(P)
1 foreach σ ∈ S do v_0(σ) ← R(σ)
2 n ← 0
3 loop
4    n ← n + 1
5    foreach σ ∈ S do
```

```
6     foreach α ∈ A(σ) do
7       g_n(σ, α) ← γ × Σ_{σ'∈S} (T(σ, α, σ') × v_{n−1}(σ'))
8     v_n(σ) ← max_{α∈A(σ)} {g_n(σ, α)}
9     π_n(s) ← arg max_{α∈A(σ)} {g_n(σ, α)}
10  if max_{σ∈S} |v_n(σ) − v_{n−1}(σ)| < ε then return π_n
```

The probabilistic planning algorithm starts by assigning value $\mathcal{R}(\sigma)$ to each state $\sigma \in \mathcal{S}$. Then, it iteratively refines these values by selecting actions that maximize the expected gains. At each iteration n, and for each state σ, the value $v_n(\sigma)$ is computed from the value $v_{n-1}(\sigma)$, that was computed at the previous iteration. It can be shown that there exists a maximum number of iterations needed to guarantee that this algorithm returns an optimal policy [Ghallab, Nau, and Traverso 2004]. However, in practical applications, the condition used to stop iteration is the following:

$$\max_{\sigma \in \mathcal{S}} |v_n(\sigma) - v_{n-1}(\sigma)| < \epsilon$$

With this condition, the algorithm guarantees that the returned policy is an ϵ-optimal policy, i.e., for each state $\sigma \in \mathcal{S}$, the expected value $v(\sigma)$ does not differ from the optimum value $v^\star(\sigma)$ by more than an arbitrarily small fixed error ϵ.

2.3 Comparison between the Approaches

In this section, we present a brief comparison between the algorithms for probabilistic planning and for nondeterministic planning, by considering the planning domain depicted in Figure 3. By analyzing the solutions that these two algorithms find for similar planning problems, we intend to indicate the advantages of each one and move toward a third alternative, which combines both of them (the resulting algorithm is presented in the next section).

The next example shows the frailties of probabilistic planning when strong policies are required.

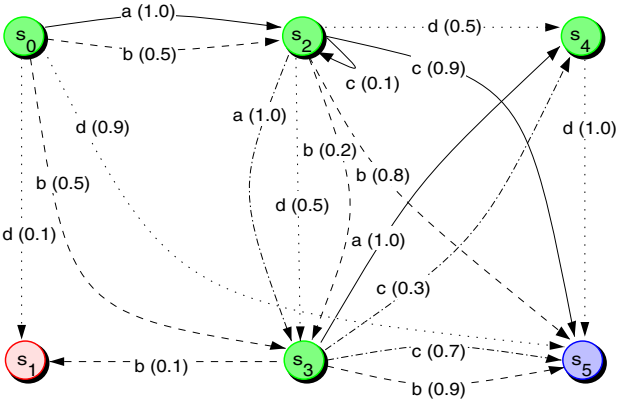


Fig. 3. A domain where actions have uncertain effects

Example 1. *Consider the planning domain $\mathcal{D}$, depicted in Figure 3, and let $\mathcal{G} = \{s_5\}$ be the set of goal states. For this problem, the algorithm* PROBABILISTICPLANNING$(\langle \mathcal{D}, \mathcal{G} \rangle)$ *returns the following policy (with $\gamma = 0.9$):*

$\pi(s_0) = d$
$\pi(s_1) = \tau$
$\pi(s_2) = c$
$\pi(s_3) = c$
$\pi(s_4) = d$
$\pi(s_5) = \tau$

This policy is an optimal expected-case solution, i.e., it has shortest execution in the expected-case. By executing action d in state s_0, we expect that in 90% of the executions the goal state can be reached with only one step. This is very efficient and, in some applications, this could be advantageous, even if 10% of the executions fail to reach the goal state. However, there are many other practical applications where failures are unacceptable. In such applications, a plan that may lead to longer executions, but necessarily reaches the goal, is preferable to a plan that in the optimistic case may reach the goal earlier, but in the pessimist case may no longer reach the goal. Clearly, the policy returned by the probabilistic algorithm is weak for state s_0, because it cannot guarantee that the goal state will be reached from this state. Therefore, if an application does not allow failures, a weak policy is inappropriate. On the other hand, if s_2 is considered as the initial state, the returned policy is a strong-cyclic solution (a better solution, because it guarantees to reach the goal state from s_2). However, due to cycles, the number of steps that a strong-cyclic policy need to reach a goal state is unbounded (e.g., in Figure 3, too many steps c could be needed until agent could leave state s_2). Therefore, if an application is critical in terms of time, a strong-cyclic policy is inappropriate. ◆

The next example illustrate the danger of excessive freedom in nondeterministic planning.

Example 2. *Consider the planning domain $\mathcal{D}$, depicted in Figure 3, and let s_0 be the initial state and $\mathcal{G} = \{s_5\}$ be the set of goal states. For this problem, the algorithm* STRONGNONDETERMINISTICPLANNING$(\langle \mathcal{D}, s_0, \mathcal{G} \rangle)$ *returns the following nondeterministic policy:*

$\pi(s_0) = a$
$\pi(s_2) = \{a, b, d\}$
$\pi(s_3) = \{a, c\}$
$\pi(s_4) = d$
$\pi(s_5) = \tau$

This policy is an optimal worst-case strong solution, i.e., it necessarily reaches the goal state after a bounded number of steps (that is minimal in the worst-case). Because in the nondeterministic model there is no preference among actions, any one of the six policies corresponding to this nondeterministic solution can be selected for execution:

$\pi_1 = \{(s_0, a), (s_2, a), (s_3, a), (s_4, d), (s_5, \tau)\}$
$\pi_2 = \{(s_0, a), (s_2, a), (s_3, c), (s_4, d), (s_5, \tau)\}$
$\pi_3 = \{(s_0, a), (s_2, b), (s_3, a), (s_4, d), (s_5, \tau)\}$
$\pi_4 = \{(s_0, a), (s_2, b), (s_3, c), (s_4, d), (s_5, \tau)\}$
$\pi_5 = \{(s_0, a), (s_2, d), (s_3, a), (s_4, d), (s_5, \tau)\}$
$\pi_6 = \{(s_0, a), (s_2, d), (s_3, c), (s_4, d), (s_5, \tau)\}$

Although an agent would prefer to select the policy π_4, which has the possibility of reaching the goal with two steps, it can even select the worst of them (π_1), which always needs exactly four steps to reach the goal state. Therefore, if an application needs an efficient strong policy, a nondeterministic strong policy is inappropriate. ◆

Remark. As we have seen, the probabilistic planning algorithm cannot guarantee to find policies that avoid failures and cycles (i.e., strong policies); conversely, the nondeterministic planning algorithm cannot guarantee to select the best strong policy. Thus, we propose a third algorithm, named *strong probabilistic planning*, that can guarantee to find an *optimal expected-case policy among those policies which are optimal in the worst-case*.

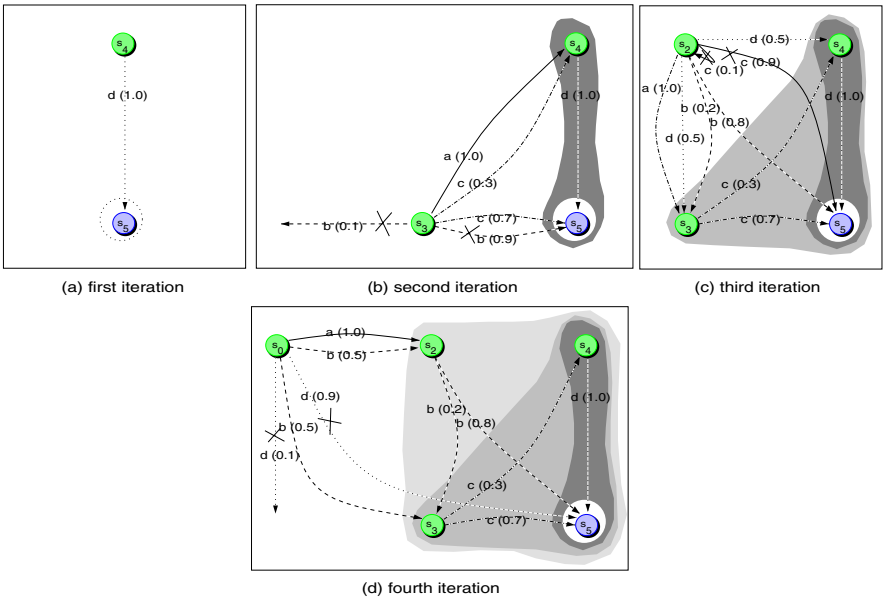
3 Strong Probabilistic Planning

The strong probabilistic planning combines two widely used approaches for planning under uncertainty. In this framework, the MC approach is used to guarantee that only optimal worst-case strong solutions can be synthesized during the planning task, while the MDP approach is used to guarantee that an optimal expected-case policy, among those that are optimal in the worst-case, is returned by the planning algorithm.

We present two versions of the algorithm for strong probabilistic planning: an *enumerative* version, where states are explicitly represented and manipulated by standard set operations, and a *symbolic* version, where states are implicitly represented by propositional formulas that can be manipulated by efficient operations on MTBDD's [Bryant 1986].

3.1 Enumerative Strong Probabilistic Planning

Given a planning problem $\mathcal{P} = \langle \mathcal{D}, s_0, \mathcal{G} \rangle$, where $\mathcal{D}$ is a probabilistic planning domain, the strong probabilistic planning algorithm starts by constructing an initial policy that maps each goal state $\sigma \in \mathcal{G}$ to the trivial action τ, and by assigning optimal expected-value 1 for each one of them. After this, in each subsequent iteration, the algorithm alternates strong preimage [Muller-Olm, Schimidt, and Steffen 1999] and optimal expected-value computations. By using the strong preimage computation, it guarantees that the synthesized policy will necessarily reach a goal state (without possibility of failure and with a bounded number of steps); and, by using the optimal expected-value computation, it guarantees that, whenever a state is mapped to more than one action by the strong preimage computation, only an optimal action will be chosen in that state. Example 3 gives some intuition about how the the strong probabilistic planning algorithm works.



(a) first iteration (b) second iteration (c) third iteration

(d) fourth iteration

Fig. 4. Strong probabilistic planning algorithm execution

Example 3. *Let $\gamma = 0.9$ and consider the planning problem $\mathcal{P} = \langle \mathcal{D}, s_0, \{s_5\} \rangle$, where $\mathcal{D}$ is the planning domain depicted in Figure 3. Initially, we have $\pi = \{(s_5, \tau)\}$ and $v(s_5) = 1$:*

- *In the first iteration (Figure 4-a), the pruned strong preimage of $\{s_5\}$ is $\{(s_4, d)\}$ and the expected gain for executing action d in state s_4 is $g(s_4, d) = \gamma \times 1.0 \times v(s_5) = 0.9$. Thus, we let $v(s_4) = 0.9$ and $\pi = \{(s_4, d), (s_5, \tau)\}$.*
- *In the second iteration (Figure 4-b), the pruned strong preimage of $\{s_4, s_5\}$, the covering set of the current policy, is $\{(s_3, a), (s_3, c)\}$. With this strong preimage computation, we can avoid action b, which could cause a failure (i.e., going from s_3 to s_1 leads the agent to a dead-end). The expected gain for the remaining actions are:*

  $g(s_3, a) = \gamma \times 1.0 \times v(s_4) = 0.81$
  $g(s_3, c) = \gamma \times (0.3 \times v(s_4) + 0.7 \times v(s_5)) = 0.87$

  With this optimal expected-case value computation, we can give preference to action c. We let $v(s_3) = 0.87$ and $\pi = \{(s_3, c), (s_4, d), (s_5, \tau)\}$. Thus, when we have to select among actions that certainly lead to the goal, we choose the one that produces the maximum expected gain.
- *In the third iteration (Figure 4-c), the pruned strong preimage of $\{s_3, s_4, s_5\}$ is $\{(s_2, a), (s_2, b), (s_2, d)\}$. Now, the strong preimage computation avoids action c, which could cause cycle. The expected gains for the other actions are:*

  $g(s_2, a) = \gamma \times (1.0 \times v(s_3)) = 0.79$
  $g(s_2, b) = \gamma \times (0.2 \times v(s_3) + 0.8 \times v(s_5)) = 0.88$
  $g(s_2, d) = \gamma \times (0.5 \times v(s_3) + 0.5 \times v(s_4)) = 0.80$

being action b the best choice in state s_2. Thus, we let $v(s_2) = 0.88$ and $\pi = \{(s_2, b), (s_3, c), (s_4, d), (s_5, \tau)\}$

- *Finally, in the last iteration (Figure 4-d), the pruned strong preimage of $\{s_2, s_3, s_4, s_5\}$ is $\{(s_0, a), (s_0, b)\}$. The action d, which could cause failure, is eliminated. The expected gains are:*

$$g(s_0, a) = \gamma \times (1.0 \times v(s_2)) = 0.789$$
$$g(s_0, b) = \gamma \times (0.5 \times v(s_3) + 0.5 \times v(s_2)) = 0.787$$

Now, action a is the best choice. Thus, we let $v(s_0) = 0.80$ and $\pi = \{(s_0, a), (s_2, b), (s_3, c), (s_4, d), (s_5, \tau)\}$. Because the initial state s_0 is covered by this policy, the strong probabilistic planning stops and returns π as solution (which corresponds to policy π_4 in the comparison section). ◆

The enumerative version. The enumerative version of the strong probabilistic planning algorithm is composed of two functions: the StrongPreimage function, that performs the strong preimage computation, and the Choose function[1], that performs the optimal expected-value computation.

Given a planning problem $\mathcal{P}$, the enumerative version of the strong probabilistic planning algorithm starts by assigning reward 1 to goal states and by building a policy that maps each one of these states to the trivial action τ. Afterwards, alternately, the algorithm computes strong preimages and optimal expected values. By using the function StrongPreimage, it guarantees that the synthesized policy necessarily reaches a goal state (in spite of nondeterminism); and by using the function Choose, it guarantees that whenever a state is mapped to more than one action (by the strong preimage function), only an action with optimal expected value can be chosen in that state.

```
StrongProbabilisticPlanning(P)
1 foreach σ ∈ G do v(σ) ← 1
2 π ← ∅
3 π' ← {(σ, τ) : σ ∈ G}
4 while π ≠ π' do
5   S ← StatesCoveredBy(π')
6   if s0 ∈ S then return π'
7   π ← π'
8   π' ← π' ∪ Choose(Prune(StrongPreimage(S), S))
9 return failure
```

```
Choose(R)
1 π ← ∅
2 foreach σ ∈ StatesCoveredBy(R) do
3   A ← {α : (σ, α) ∈ R}
4   foreach α ∈ A do
5     g(σ, α) ← γ × Σ_{σ'∈T(σ,α)} (T(σ, α, σ') × v(σ'))
```

[1] Because all paths in a strong policy have a bounded number of steps (finite horizon), a discount factor is no longer necessary to guarantee convergence; however, it is still necessary to force the agent to give preference to shortest paths.

6 $v(\sigma) \leftarrow \max_{\alpha \in A}\ g(\sigma, \alpha)$
7 $\pi \leftarrow \pi \cup \{(\sigma, \arg\max_{\alpha \in A}\ g(\sigma, \alpha))\}$
8 `return` π

The function CHOOSE implements the optimality principle through backward induction. This is possible because the combined use of the functions STRONGPREIMAGE and PRUNE avoids cycles, allowing a topological sorting of the states covered by the policy (Figure 5). By processing states in reverse topological order, when the value of a state is computed, the value of each one of its possible successors is already known. This can reduce the number of state updates.

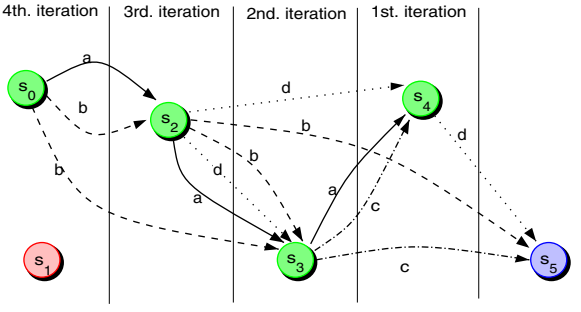


Fig. 5. Topological sorting of the domain's states

The following theorems are the main results because they prove that backward induction works for our model.

Theorem 1. *If a probabilistic planning problem $\mathcal{P}$ has a strong solution, the algorithm* STRONGPROBABILISTICPLANNING *returns an optimal worst-case strong policy for $\mathcal{P}$.*

Proof. We denote by π_i the policy built in the i-th iteration of the algorithm. By definition, a state $\sigma \in \mathcal{S}$ is covered by π_0 if and only if σ is a goal state; thus, π_0 covers all states from which, in the worst case, there is a path of length 0 to a goal state. In the first iteration, if the initial state s_0 is covered by π_0, clearly, the algorithm returns an optimal worst-case policy for $\mathcal{P}$. Otherwise, the pruned strong preimage of the set S_0 of states covered by π_0 is computed. For each pair $(\sigma, \alpha) \in$ PRUNE(STRONGPREIMAGE$(S_0), S_0)$), all α-successors of σ are goal states, independently of the chosen actions; thus, the policy $\pi_1 := \pi_0 \cup$ CHOOSE(PRUNE(STRONGPREIMAGE$(S_0), S_0)))$ covers all states from which, in the worst case, there is a path of length 1 to a goal state. By the inductive hypothesis, for $j < i$, policy π_j covers all states from which, in the worst case, there exists a path of length j to a goal state. Therefore, in the i-th iteration, if the initial state s_0 is covered by π_{i-1}, the algorithm returns an optimal worst-case policy for $\mathcal{P}$. Otherwise, the pruned strong preimage of the set S_{i-1} of states covered by π_i is computed. If $(\sigma, \alpha) \in$ PRUNE(STRONGPREIMAGE$(S_{i-1}), S_{i-1})$), then at least one α-successor of σ takes, in the worst case, $i - 1$ steps to reach a goal state (otherwise the state σ would have been covered by policy π_{i-1} and, thus, been

pruned). Therefore, independently of the chosen actions, the policy $\pi_i := \pi_{i-1} \cup$ CHOOSE(PRUNE(STRONGPREIMAGE(S_{i-1}), S_{i-1}))) covers all states from which, in the worst case, there is a path of (optimal) length i to a goal state. ◆

Theorem 2. *The optimal worst-case strong policy returned by algorithm* STRONGPROBABILISTICPLANNING *is optimal in the expected-case.*

The expected-case optimality of the policy returned by the algorithm STRONGPROBABILISTICPLANNING is derived from the fact that function CHOOSE uses the optimality principle [Bellman 1957] to choose the best action for each state covered by this policy.

3.2 Symbolic Strong Probabilistic Planning

The basic idea underlying the symbolic version of the strong probabilistic planning algorithm is to represent states as sets of propositions and to consistently work with propositional formulas that characterize sets of states. In order to do this, a new definition of planning domain is needed:

Definition 5. *A symbolic probabilistic planning domain is a tuple* $\mathcal{D} = \langle \mathbb{P}, \mathcal{S}, \mathcal{A}, \mathcal{L}, \mathcal{T} \rangle$, *where:*

- $\mathbb{P}$ *is a finite nonempty set of atomic propositions;*
- $\mathcal{S}$ *is a finite nonempty set of states;*
- $\mathcal{A}$ *is a finite nonempty set of actions;*
- $\mathcal{L} : \mathcal{S} \mapsto 2^{\mathbb{P}}$ *is a state labeling function;*
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ *is a state transition function.* ◆

Each atomic proposition $p \in \mathbb{P}$ denotes a state property. The set of atomic propositions which are satisfied in a state $\sigma \in \mathcal{S}$ is denoted by $\mathcal{L}(\sigma)$. The *intension* of a propositional formula φ in $\mathcal{D}$, denoted by $[\![\varphi]\!]_{\mathcal{D}}$, is the set of states in $\mathcal{D}$ which satisfies φ. Formally, we have[2]:

- $[\![\varphi]\!] = \{\sigma \in \mathcal{S} : \varphi \in \mathcal{L}(\sigma)\}$ if $\varphi \in \mathbb{P}$
- $[\![\neg\varphi]\!] = \mathcal{S} \setminus [\![\varphi]\!]$
- $[\![\varphi \wedge \varphi']\!] = [\![\varphi]\!] \cap [\![\varphi']\!]$
- $[\![\varphi \vee \varphi']\!] = [\![\varphi]\!] \cup [\![\varphi']\!]$

Furthermore, we assume that $\top \in \mathcal{L}(\sigma)$, for all state $\sigma \in \mathcal{S}$. Therefore, it follows that $[\![\top]\!] = \mathcal{S}$.

The trivial action $\tau \in \mathcal{A}$ and the transition function $\mathcal{T}$ are defined as in the pure probabilistic case.

Definition 6. *A symbolic probabilistic planning problem is a tuple* $\mathcal{P} = \langle \mathcal{D}, s_0, (\varphi, \varphi') \rangle$, *where:*

- $\mathcal{D}$ *is a symbolic probabilistic planning domain;*
- $s_0 \in \mathcal{S}$ *is an initial state;*
- (φ, φ') *is an extended reachability goal.* ◆

[2] For the sake of simplicity, we omit subscript $\mathcal{D}$ in $[\![.]\!]$.

An *extended reachability goal* [Pereira and Barros 2008] is a pair of logical formulas (φ, φ'): the *preservation condition* φ specifies a property that should hold in each state visited through the path to a goal state (excepting the goal state); and the *achievement condition* φ' specifies a property that should hold in all goal states, *i.e.*, $\mathcal{G} = [\![\varphi']\!]_{\mathcal{D}}$.

Extended goals (Pistore, Bettin, and Traverso 2001; Lago, Pistore, and Traverso 2002; Pereira and Barros 2008) represent an improvement on the expressiveness of the reachability planning framework. By using such goals, besides defining acceptable final states, we can also establish preference among possible intermediate states. Note that a reward function can have the same expressiveness of extended goals; however, extended goals are high level specifications.

An example of a symbolic probabilistic domain is depicted in Figure 6. The shadowed states are the ones that can be covered by a policy for the extended reachability goal $(\neg q, p \wedge q \wedge r)$, which specify that the agent should preserve property $\neg q$ (equivalently, avoid property q), until reaching a state where the three properties p, q and r can be satisfied. Other examples of useful extended reachability goals are:

- $(\top, r)$: to achieve property r;
- (p, r): to achieve property r, by preserving property p;
- $(\neg q, r)$: to achieve property r, by avoiding property q;
- $(p \wedge \neg q, r)$: to achieve r, by preserving p and avoiding q.

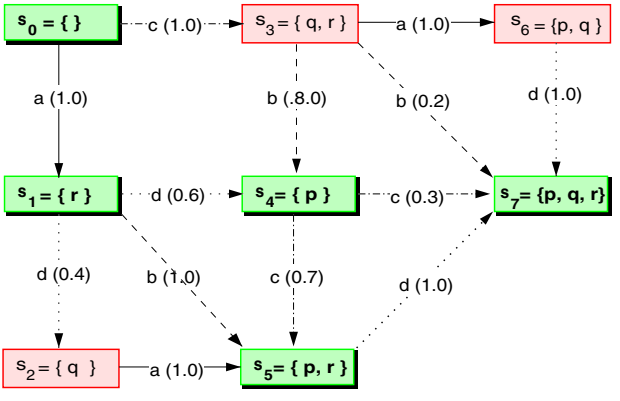


Fig. 6. A symbolic probabilistic planning domain

The symbolic version. The symbolic version for the algorithm for extended reachability goals is very similar to the enumerative one. The main difference is on the "intensional" representation of set of states and on the definition of the prune function, which is defined as following:

PRUNE(R, S, φ)
1 `return` $\{(\sigma, \alpha) \in R : \sigma \in [\![\varphi]\!]_{\mathcal{D}}$ and $\sigma \in S\}$

Given the strong preimage R of a set of states S, as well as a preserving condition φ, the function PRUNE selects from R all pairs (σ, α), such that state σ has property φ

and it was not yet mapped to another action in a previous iteration. By proceeding in this way, the prune function avoids all intermediate states which does not satisfy the preserving condition φ.

The remainder of the planning algorithm is as following:

```
StrongProbabilisticPlanning(P)
1 foreach σ ∈ [[φ']]_D do v(σ) ← 1
2 π ← ∅
3 π' ← {(σ, τ) : σ ∈ [[φ']]_D}
4 while π ≠ π' do
5   S ← StatesCoveredBy(π')
6   if s_0 ∈ S then return π'
7   π ← π'
8   π' ← π' ∪ Choose(Prune(StrongPreimage(S), S))
9 return failure

Choose(R)
1 π ← ∅
2 foreach σ ∈ StatesCoveredBy(R) do
3   A ← {α : (σ, α) ∈ R}
4   foreach α ∈ A do
5     g(σ, α) ← γ × Σ_{σ'∈T(σ,α)} (T(σ, α, σ') × v(σ'))
6   v(σ) ← max_{α∈A} g(σ, α)
7   π ← π ∪ {(σ, arg max_{α∈A} g(σ, α))}
8 return π
```

The following theorems prove that backward induction also works for the symbolic version of our model.

Theorem 3. *If a symbolic probabilistic planning problem $\mathcal{P}$ has a strong solution, the symbolic version of algorithm* StrongProbabilisticPlanning *returns an optimal worst-case strong policy for $\mathcal{P}$.*

Theorem 4. *The optimal worst-case strong policy returned by the symbolic version of algorithm* StrongProbabilisticPlanning *is optimal in the expected-case.*

The proofs to these theorems are straightforward from proof of Theorem 1. Noticing that, besides pruning the states already covered by the policy under construction, the symbolic version of the function Prune also prunes states that do not satisfy the extended reachability goal.

4 A Note about Implementation

The policies for the examples in this paper were synthesized by programs that we have implemented. The algorithm ProbabilisticPlanning was implemented in Java, while the other two – StrongNondeterministicPlanning and StrongProbabilisticPlanning – were implemented in Prolog. Because the comparison of techniques does not take into account efficiency issues, the use of different programming languages for the implementations does not affect our analysis.

5 Conclusion

In this paper we have identified, and solved, the problem of *strong probabilistic planning*. In essence, this is a situation with features of nondeterministic and probabilistic planning: requirements on the planning goals mix worst-case and expected-case analysis, and actions with (uniform) costs and (Markovian) probabilities associated with them.

Our main contribution is to show that the resulting problem can be tackled efficiently by using backward induction (with minimum number of state updates), thus producing the enumerative and symbolic versions of the StrongProbabilisticPlanner algorithm. While Theorems 1 and 2 deal with straightforward reachability goals, Theorems 3 and 4 show that our techniques can be applied in much greater generality to extended reachability goals. With such goals we can also impose constraints on states visited during policy execution. Hence the symbolic framework is more expressive than the enumerative one. As expressiveness increases planner usability, the symbolic framework seems to be more appropriate for practical planning applications.

The desire to combine features of nondeterministic and probabilistic planning have led us to develop a perspective for planning problems that integrates these features coherently, as we feel that current literature treats these varieties of planning as too isolated islands. We have tried to convey some of this perspective in the second section of this paper; we hope that the resulting blend improves understanding of this multifaceted area.

We should still emphasize that we can found in the literature algorithms that are capable of synthesizing weak solutions (e.g., *value-iteration* [Puterman 1994]), as well as strong-solutions (e.g. RTDP [Bertsekas and Tsitsiklis 1991]), for probabilistic planning problems modeled as MDPs. However, to the best of our knowledge, the hybrid framework proposed in this paper is the first one that is capable of guarantee the synthesis of strong solutions for such kind of probabilistic planning problems (the work in [Mausam, Bertoli, and Weld2007] does not offer this guarantee).

References

Altman, E.: Constrained Markov Decision Processes. Chapman & Hall / CRC, Florida (1999)

Bellman, R.E.: Dynamic Programming. Princeton University Press, USA (1957)

Bertsekas, D.P., Tsitsiklis, J.N.: An analysis of stochastic shortest path problems. Math. Oper. Res. 16(3), 580–595 (1991)

Boutilier, C., Dean, T., Hanks, S.: Decision-theoretic planning: Structural assumptions and computational leverage. Journal of Artificial Intelligence Research 11, 1–94 (1999)

Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers 35(8), 677–691 (1986)

Cimatti, A., Giunchiglia, F., Giunchiglia, E., Traverso, P.: Planning via model checking: A decision procedure for $\mathcal{AR}$. In: ECP, pp. 130–142 (1997)

Cimatti, A., Roveri, M., Traverso, P.: Strong planning in non-deterministic domains via model checking. In: Artificial Intelligence Planning Systems, pp. 36–43 (1998)

Daniele, M., Traverso, P., Vardi, M.Y.: Strong cyclic planning revisited. In: Biundo, S., Fox, M. (eds.) ECP 1999. LNCS, vol. 1809, pp. 35–48. Springer, Heidelberg (2000)

Dolgov, D.A., Durfee, E.H.: Stationary deterministic policies for constrained MDPs with multiple rewards, costs, and discount factors. In: IJCAI, pp. 1326–1331 (2005)

Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practice. Morgan Kaufmann Publishers Inc., USA (2004)

Giunchiglia, F., Traverso, P.: Planning as model checking. In: ECP, pp. 1–20 (1999)

Lago, U.D., Pistore, M., Traverso, P.: Planning with a language for extended goals. In: Eighteenth national conference on Artificial intelligence, pp. 447–454. American Association for Artificial Intelligence, Menlo Park (2002)

Mausam Bertoli, P., Weld, D.S.: A hybridized planner for stochastic domains. In: Veloso, M.M. (ed.) IJCAI, pp. 1972–1978 (2007)

Muller-Olm, M., Schimidt, D., Steffen, B.: Model checking: A tutorial introduction. In: Cortesi, A., Filé, G. (eds.) SAS 1999. LNCS, vol. 1694, pp. 330–354. Springer, Heidelberg (1999)

Pereira, S.L., Barros, L.N.: A logic-based agent that plans for extended reachability goals. Autonomous Agents and Multi-Agent Systems 9034, 327–344 (2008)

Pistore, M., Bettin, R., Traverso, P.: Symbolic techniques for planning with extended goals in non-deterministic domains (2001)

Puterman, M.L.: Markov Decision Processes—Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., Chichester (1994)

Parametric Algorithms for Cyclic Scheduling Problems with Applications to Robotics

Vladimir Kats[1] and Eugene Levner[2]

[1] Institute for Industrial Mathematics, Beer-Sheva, Israel
vkats@iimath.com
[2] Holon Institute of Technology, Holon and Bar-Ilan University, Ramat Gan, Israel
levner@hit.ac.il

Abstract. The paper considers cyclic scheduling problems arising in planning of robotic flowlines and logistics. The problems are reduced to finding the minimum (or maximum) cost-to-time ratio in doubly weighted graphs. New combinatorial algorithms are developed extending the cyclic project scheduling algorithm by Romanovskii (1967) and the minimum-cost-to-time-ratio algorithm by Dantzig, Blattner and Rao (1967). Whereas Romanovskii's and Dantzig-Blattner-Rao's algorithms are restricted to solving problems with fixed numerical data, the new algorithms can treat input data that are varied at prescribed intervals. Several special cases are solved in strongly polynomial time.

Keywords: Cyclic scheduling, robotic scheduling, polynomial time algorithm, intersection of polygons.

1 Introduction

This work considers periodically repeated PERT/CPM projects in which a large number of interrelated operations is performed simultaneously by parallel machines. The problem is to schedule the projects so as to maximize throughput by minimizing cycle time. We call this problem the *cyclic project scheduling problem.* There are many examples of its applications in different industries (see, e.g. Lei 1993, Lee and Posner 1997), robotics (Crama et al. 2000, Dawande et al. 2007), parallel computing (Hanen and Munier 1995), and transportation and logistics (Dantzig et al. 1967.) The advantages of cyclic scheduling over non-cyclic scheduling in manufacturing and transportation systems include more efficient material handling, better resource utilization and simpler control.

We study several variants of the cyclic scheduling problems and discuss computational methods starting with the seminal algorithms developed by Dantzig, Blattner and Rao (1967) and Romanovskii (1967). Whereas Romanovskii's and Dantzig-Blattner-Rao's algorithms are restricted to solving problems with fixed numerical data, we develop extended algorithms that can treat input data varying in prescribed intervals. Several special cases are solved in strongly polynomial time.

Along with standard cyclic schedules, called 1-*cyclic*, in this paper we also consider the so-called 2-*cyclic schedules* for the robotic flowlines with interval-valued processing times. We develop a polynomial time algorithm for solving the 2-cyclic

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 653–663, 2008.

problem, exploiting for this aim Meggido's (1983) or Dyer's (1984) fast algorithms for two-variable linear programming as a sub-procedure.

The paper is organized as follows. In the next section we formulate the cyclic problems. In Section 3 we discuss the relationships between the cyclic project scheduling problems and problems of finding extremal cycles in graphs. In Section 4 we provide new polynomial time algorithms extending earlier Romanovskii's and Dantzig-Blattner-Rao's algorithms. We develop also the strongly polynomial time algorithm for finding optimal 2-cyclic schedules. Section 5 concludes the paper.

2 Problem Formulation

2.1 The Cyclic Project Scheduling Problem

An early approach to the cyclic project scheduling problem was introduced by Romanovskii (1967). He thoroughly investigated its graph-theoretic model. However, this remarkable work did not receive much recognition at that time. But later this graph model was re-discovered, revised and extended in a series of works by Hanen and Munier (1995), Lee and Posner (1997), Kampmeyer (2006), and many others.

There is a set S of n partially ordered operations, called *generic operations*, to be performed in parallel by machines or operators. As in the classical PERT/CPM problem formulation, each operation is performed by a dedicated machine and it is assumed that there are sufficiently many such machines, so the question of scheduling of operations on machines vanishes. Each operation i has a fixed processing time $p_i > 0$. Consider a periodic process in which each operation must be performed periodically, infinitely many times, with the same period T.

For each operation i, let $<i, k>$ denote the kth *execution* (or *repetition*) of operation i in the periodic process where k is any positive integer. A *schedule* is an assignment of a starting time $t(i, k)$ to each execution (i, k). A schedule $\{t(i, k)\}$ satisfying the precedence relations is called *feasible.* A feasible schedule is called *cyclic* with *cycle time* T if $t(i, k) = t(i, 1) + (k\text{-}1)T$, for all integer $k \geq 1$ and all $i \in S$.

The generic operations and the precedence relations between them are expressed in a *generic graph* G. Each operation is depicted as a node. If operation i precedes operation j the corresponding *arc* (i,j) is introduced. Each arc of this graph is associated two weights, one to denote its processing times or time gaps between the operations (called the *arc length* L_{ij}) and the other to denote the distance between operations occurring within different periods measured in the number of periods (called the *arc height* H_{ij}). For any pair of operations i and j, with given length L_{ij} and height H_{ij}, the following relations are given: $t(i, k) + L_{ij} \leq t(j, k + H_{ij})$, for all $k \geq 1$.

The problem is to find a periodic schedule, i.e., the starting time $t(i, k)$ of each operation, yielding a minimum cycle time $T = T^*$.

2.2 The Cyclic Robotic Flowshop Scheduling Problem

A robotic production line has two types of resources, namely workstations and robots. Consider a flow-line having M workstations, say $S_1, \ldots, S_M$. Let S_0 and S_{M+1} denote the input and output stations, respectively. The line processes identical parts. The order of operations is represented by the processing sequence $\mathbf{S} = (S_0, S_1, \ldots, S_M, S_{M+1})$, where

the kth component indicates that the kth operation is performed in workstation S_k, k = 0, …, M+1, i.e., each part starts at the input station S_0, passes successively through the stations S_1, …, S_M and is finally unloaded at the output station S_{M+1}.

A robot repeatedly transports parts between the workstations and load/unload them. Such a repetition of robot moves is called a *robot cycle* and its duration the *cycle time*, T. The objective is to maximize the throughput rate. The throughput rate will be maximum when T is minimum. We assume that the robot's route is given.

A robot cyclic route is denoted by **R**. Without loss of generality, it starts with the input station S_0. Since the loaded robot always moves from a workstation $S_{R(k)}$ to the next one in the sequence **S,** namely $S_{R(k)+1}$, the robot route **R** can be presented compactly as **R** = ($S_{R(0)} = 0$, $S_{R(1)}$, $S_{R(2)}$, …, $S_{R(M)}$), or, simply, **R** = ([0], [1], …., [M]) using [k] instead of $S_{R(k)}$. The sequence **R** is a cyclic permutation of (0,1, …, M).

The scheduling problem must satisfy the following conditions:

- Upon arrival at workstation S_k, a part is loaded and stays there for p_k time units, where $L_k \le p_k \le U_k$, L_k and U_k are given constants, k = 1, …, M.
- The loaded robot requires d_k time units to move a part from station S_k to S_{k+1}, k = 1, …, M.
- The unloaded robot requires r_{ij} time units to run from S_i to S_j, $2 \le i \le M+1$, $0 \le j \le M$.

A *no-wait condition* requires that after a part is processed in a workstation, it must be unloaded and moved immediately to the next workstation in **S** and then loaded and processed on it without pause. If this condition is not imposed, a part may stay in a workstation after it is finished. The suggested graph approach permits to treat with the same computational complexity both the "no-wait" and "no-no-wait" models.

The variables in the problem are:

- p_k = the duration of the kth operation in the processing sequence **S**,
- t_k = the completion time of the kth operation in the processing sequence **S** for the part which is started at moment 0, where k = 1, …, M, and
- T = the cycle time.

Given the processing set **S**, the robot's route **R**, real numbers d_k, r_{ij}, L_k and U_k (k = 1, …, M, i = 2, …, M+1, j = 0, …, M), a *feasible schedule* is a set of operation completion times t_k and a set of operation durations p_k (k = 1, …, M) lying within the prescribed intervals [L_k, U_k] and providing that the empty robot has sufficient time to travel between the workstations. The cycle time of a feasible schedule is called the *feasible cycle time*.

The scheduling problem is to find the interval $\boldsymbol{T}$ of all the feasible cycle time values. The minimal T from $\boldsymbol{T}$ yields the minimum cycle time.

3 Graph Reformulation of the Cyclic Scheduling Problems

Consider the cyclic robotic flowshop problem. Let us denote by **A** the set of part-processing operations in **S** that being started in some robot cycle **R** are finished in the same cycle and by **B** those operations in **S** that being started in a cycle **R** are finished in the next cycle. This information is obtained by examining sequences **S** and **R**.

Then the cyclic scheduling problem considered can be re-formulated as the following parametric linear programming problem:

$$\text{Problem LP:} \quad \text{Find } T^* = \text{minimum } T \tag{1}$$

subject to

$$L_k \le p_k \le U_k,\ k = 1, \ldots, M, \tag{2}$$

$$p_k = t_k - t_{k-1} - d_{k-1}, \text{ if } k \in \mathbf{A}, \tag{3}$$

$$p_k = T + t_k - t_{k-1} - d_{k-1}, \text{ if } k \in \mathbf{B}, \tag{4}$$

$$t_{[k]} + d_{[k]} + r_{[k]+1,[k+1]} \le t_{[k+1]}, \text{ for } k = 1,2, ..., M, \tag{5}$$

$$t_1 = 0,\ t_{[M+1]} = T. \tag{6}$$

Relation (2) describes the two-sided constraints on the processing times, (3) and (4) express the relationships between processing times and completion times in the no-wait case, depending on whether operation k is from set **A** or **B**. Inequalities (5) guarantee that the unloaded robot has enough time to arrive at each machine ($W_{[k+1]}$) before the corresponding operation on this machine is finished.

We construct graph $G_1 = (N_1, A_1)$ with node set N_1 and arc set A_1. Each *node* in N_1 corresponds to the completion of the operations from **S** plus one more (dummy) node, $f = [M+1]$, which corresponds to the additional variable $t_{[m+1]} = T$. The node corresponding to the completion of the loading operation at S_0, is chosen as the *source*.

In order to define the arc set, we substitute (3) and (4) into (2). Then constraints of the form $L_k \le t_k - t_{k-1} - d_{k-1}$ and $L_k \le T + t_k - t_{k-1} - d_{k-1}$ are represented by arcs leading from node k - 1 to node k and having the *generalized length* $L_k + d_{k-1}$ and $L_k - T + d_{k-1}$, respectively. The former arcs are called *forward arcs*, and the latter, whose length depends on T, are called *backward arcs*. Similarly, all the other constraints are represented by the arcs such that each arc $e = (i, j) \in A$ is assigned "*length*" of the form $W(e) = L(e) - TH(e$, where $L(e)$ are real numbers (either positive, negative or zero) and $H(e) \in \{-1, 0, 1\}$. So, each arc is associated with two values, L and H. The first value L is called a *length* and the second value H a *height*. The graph model for the project scheduling problem of Section 2.1 is constructed in a similar way.

There are two basic approaches to solving these cyclic scheduling problems reformulated in graph terms. The first one associates each arc e with two *weights*, L(e) and $H(e)$, and exploits the fact that the minimum cycle time T^* equals the maximum cycle length-to-height ratio, $\max_C L(C)/H(C)$, where C is a simple cycle in the doubly-weighted graph. This scheme will be utilized in Section 4.2. The second approach considers graph $G_1 = (N_1, A_1)$ with arc-assigned *generalized lengths* $W(e) = L(e) - TH(e)$, parametrically depending on T, e.g., $W(e) = 23 - T$; its goal is to find the *parametric critical path* in the graph. This scheme will be used in Section 4.3.

4 Fast Cyclic Scheduling Algorithms

4.1 Comparison of Two Classical Algorithms

In this subsection, we compare two graph algorithms, by Romanovskii (1967) and Dantzig et al. (1967). We start with a brief description of the Romanovskii algorithm

(1967) for finding the maximum length-to-height cycle ratio in a graph G. Its main idea will be used for solving a general problem with interval-valued data.

Step 1. Select an arbitrary simple cycle C in G as a *starting cycle* and put $\chi_C = \Sigma e \in CL(e)/\Sigma e \in CH(e)$; $L'(e) = L(e) - \chi_C H(e)$. [Graph $G(\chi_C)$ with *distances* $L'(e)$, where χ_C is fixed, is called *revised*. As long as the problem in the revised graph $G(\chi_C)$ with distances $L'(e)$ has a positive circuit, the value of ratio χ_C can be further increased].

Step 2. Select an arbitrary node i_0 on the cycle C and assign potential $\varphi(i_0) = 0$. Construct a subgraph G' of G such that from each node exactly one arc goes out and the only circuit of which is C. (All the nodes of C succeed all the other nodes of G'). For each arc $e = (i, j)$ in G' leading from node i to node j, put

$$\varphi(i) = \varphi(j) + L'(e). \tag{7}$$

Step 3. For each node i and all $e = (i, j) \in U(i)$, where $U(i)$ is the set of arcs of G going out of i, the following inequality is checked:

$$\varphi(i) \geq \varphi(j) + L'(e). \tag{8}$$

If for some i and $e = (i, j) \in U(i)$ this inequality is not satisfied, then two cases are possible: (1) j does not precede i in G', and (2) j precedes i in G'. In the first case, the arc outgoing from i in subgraph G' must be replaced by the arc e, the function φ recalculated by (7), and the checking of (8) repeated. In the second case, the path $i \rightarrow j \rightarrow \ldots \rightarrow i$ represents a (positive-length) cycle with a ratio greater than χ_C. This cycle is taken as a new starting cycle C, and the process is reiterated.

Finally, one obtains a function φ satisfying (7) and (8) and, consequently, maximizing the length-to-height ratio.

The Dantzig-Blattner-Rao algorithm (DBR, 1967) solves the complementary minimum lenth-to-height cycle ratio in a graph. It has a similar structure. Comparisons of the structures of two basic cyclic algorithms are given in Table 1.

Table 1. Comparisons of two cyclic algorithms

Algorithm	*Objective*	*Starting point*	*Strategy*	*Stopping rule*
Romanovskii (1967)	Maximum cycle ratio	Choose a small parameter χ and search for a positive cycle	Starting with an infeasible parameter value χ, increase it until a tree of critical paths is obtained	Increase χ until no positive cycle can be found
Dantzig-Blattner-Rao (1967)	Minimum cycle ratio	Choose a large parameter χ and search for a negative cycle	Starting with an infeasible parameter value χ, decrease it until a tree of shortest paths is obtained	Decrease χ until no negative cycle can be found

Remark. Parameter χ in both the algorithms is the length-to-height ratio value.We call a parameter value *feasible* if the revised graph has a tree of finite shortest (longest) paths for the minimum (maximum) ratio problem. Otherwise, a parameter value is called *infeasible*.

4.2 Fast Scheduling Algorithms for Interval-Valued Data

Both the algorithms of DBR and Romanovskii, being applicable for numerical non-negative lengths and heights, may not be applied for solving problems with *interval-valued* input data, nor problems with bounded time gaps between operations. Indeed, in this case, the stopping rules summarized in Table 1 become invalid. This happens because the infeasible parameter values, or i.p.v. in short, are located in two disjoint semi-infinite intervals. Due to this reason, if, for a current parameter value, the "graph-has-no-negative-circuits" situation happens in the minimum ratio problem (and the "graph-has-no-positive-circuits" situation happens in the maximum ratio problem), then it does not necessarily follow that the optimal solution of the problem is found. Although the original DBR and Romanovskii algorithms are invalid for solving the ratio problems with interval-valued data, they both can be modified.

The extended algorithm for the minimum ratio problem works as follows. Given an initial graph G and a current value of ρ, the *revised graph* $G(\rho)$, with modified arc distances $d(i, j) = c(i, j) - \rho t(i, j)$, is defined as in the DBR algorithm. The minimum ratio ρ* is the largest value of ρ for which $G(\rho)$ has no negative-distance cycles, as in the case of non-negative data. The algorithm starts with an *arbitrary* value of ρ and checks whether or not the graph $G(\rho)$ has a negative-distance circuit C.

In the first case (i.e., when a negative-distance circuit is found), the algorithm either (i) increases ρ if $H_C < 0$, or (ii) decreases ρ if $H_C > 0$, until no negative-distance circuit can be found. In case (i), after discovering that there is no negative circuit, the algorithm builds a set of shortest-distance trees and increases ρ from one tree to another until in some iteration a negative-distance circuit appears in the current tree. Then the latest current ρ value, for which there is no negative distance circuit, is the desired value ρ*. In case (ii), the maximum ρ value for which there is no negative distance circuit in $G(\rho)$ is the desired minimum ratio ρ*. In the second case (i.e., when $G(\rho)$ has no negative circuits), the algorithm constructs the shortest distance trees in the same way as it does under subcase (i) of the first case.

In a similar way, the Romanovskii algorithm for solving the maximum circuit ratio problem with positive input data can be extended to treat the case with real (not necessarily integer) data of any sign. The maximum ratio ρ* is the smallest value of ρ for which $G(\rho)$ has no positive-distance cycles. The algorithm starts with an arbitrary parameter value ρ and searches for a *positive-distance* circuit in graph $G(\rho)$.

Two cases are possible: either (1) graph $G(\rho)$ has such a circuit C, or (2) there is no circuit of positive distance in $G(\rho)$. In case (1), two sub-cases are possible: (i) the total height H_C of the found circuit C is positive, or (ii) the height H_C is negative. In case 1(i), the algorithm increases ρ until no positive-distance circuit can be found; the final ρ value obtained is the desired ρ*. In case 1(ii), the algorithm decreases ρ until no positive-distance circuit can be found. The found ρ value is feasible but not minimally possible among feasible values. Then the algorithm builds a sequence of longest-distance trees such that feasible ρ values decrease from one tree to another until a positive-distance circuit appears again in the current tree. The last found ρ value is the desired ρ*. In the second case, the algorithm builds and re-constructs the longest distance trees in the same way as it does under sub-case (ii) of the first case.

The complexity of the above algorithms for arbitrary input data is an open question. For the special case concerning the robotic scheduling problem, with M machines and

heights -1, 0, or 1, the algorithms run in $O(M^2 \log M)$ time. For a more general case in which the second weights on arcs are integers with values from the interval $[-H_{\max}, H_{\max}]$, the algorithm complexity is $O(bH_{\max} M \log M)$, where $b \le M$.

4.3 The 2-Cyclic Scheduling Algorithm for Interval-Valued Data

A *m-cyclic schedule* is a sequence of tasks executed repeatedly so that each task is performed m times during each cycle. Many researchers have noticed and experimentally verified that the mean cycle time in 2-cyclic schedules can be better than in the optimal 1-cyclic schedules ([1-3, 13, 14, 19].) However, in comparison with 1-cyclic schedules finding optimal 2-cyclic schedules is more complicated. We provide a strongly polynomial-time algorithm minimizing the mean cycle time in the case when the input data are lying in prescribed intervals, its complexity being $O(M^{11})$.

In the 2-cyclic schedules, exactly two parts enter and two parts leave the line during each cycle. The (identical) parts are loaded into the line at time ... $-kT$, $-kT + T_1$, ..., $-2T$, $-2T+T_1$, $-T$, $-T+T_1$, 0, T_1, T, $T+T_1$, $2T$, ..., $kT+T_1$, $(k+1)T$, where $T_1 < T$ and k is an integer. The parts loaded at time kT (respectively, at time $kT + T_1$), $k = 0,\pm1,\pm2, \ldots$ are called *the parts of Class* 1 (respectively, *of Class* 2). T is called the *cycle time,* T_1 is called the *semi-cycle time*; these are the unknowns to be determined.

The order of operations on M machines is represented by the processing sequence $\mathbf{S} = (S_0, S_1, \ldots, S_M, S_{M+1})$, where the kth component indicates that the kth operation is performed in station S_k, k=0, …, M+1. The given robot route will be $m(M+1)$ long, e.g. in the case of $m = 2$, $\mathbf{R} = ([0], [1], \ldots., [M], [M+1], \ldots., [2M+2])$.

Given the processing sequence **S**, robot's route **R**, real numbers d_j, r_{ij}, L_k and U_k ($k = 1, \ldots, M$, $i = 2, \ldots, M+1$, $j = 0, \ldots, M$), a *feasible schedule* is a set of operation completion times t_k for Class 1 (and, respectively, t'_k for Class 2), and a set of operation durations p_k ($k = 1, \ldots, M$) lying within the prescribed intervals $[L_k, U_k]$ and providing that the empty robot has sufficient time to travel between the workstations. The scheduling problem is to find the *feasible schedule* such that T is the minimum.

For a given periodically repeated robot route $\mathbf{R} = ([0], [1], \ldots., [M], [M+1], \ldots., [2M+1])$, the completion times t_k and t'_k satisfy the following chain of inequalities

$$0 = t^*_0 < t^*_{[1]} < t^*_{[2]} \ldots < t^*_{[2M+1]} < t^*_{[2M+2]} = T \tag{9}$$

where $t^*_{[q]}$ denotes either t_k or t'_k, namely, $t^*_{[q]} = t_k$ if the part at machine k is of Class 1 and $t^*_{[q]} = t'_k$, if the part is of Class 2. This information is given by sequence **R**.

Then, taking into account the chain of inequalities (9), the scheduling problem in interval $[0,T)$ can be formulated as the following linear program:

$$\text{Problem P:} \quad \text{Find } T^* = \text{minimum } T \tag{10}$$

subject to

$$L_k \le p_k \le U_k, \tag{11}$$

$$p_k = t_k - t_{k-1} - d_{k-1}, \text{ if } t_k > t_{k-1} \text{ in (9)}, \tag{12}$$

$$p_k = T + t_k - t_{k-1} - d_{k-1}, \text{ if } t_k < t_{k-1} \text{ in (9)}, \tag{13}$$

$$t^*_{[q]} + d_{[q]} + r_{[q]+1,[q+1]} \le t^*_{[q+1]}, \tag{14}$$

$$t'_k = t_k + T_1, \text{ if } t_k < t'_k \text{ in (9)}, \tag{15}$$

$$t'_k = t_k + T_1 - T, \text{ if } t'_k < t_k \text{ in (9)}, \tag{16}$$

$$t_0 = 0,\ t_{[2M+2]} = T. \tag{17}$$

where $k = 1, \ldots, M;\ q = 0, 1,2, ..., 2M+1$.

Relation (11) describes the constraints on the processing times, (12) and (13) express the relationships between processing times and completion times in the no-wait case, depending on the mutual position of t_k and t_{k-1} in (9). Inequalities (14) guarantee that the robot has enough time to arrive at each machine ($S_{[q+1]}$) before the corresponding processing operation on this machine is finished. The equations (15) and (16) describe relations between variables of Class 1 and Class 2, depending on their mutual position in (9); (17) gives boundary conditions. Replacing variables p_k and t'_k by their expressions (12)-(13) and (15)-(16), correspondingly, we receive a linear program with variables T, T_1, t_1, t_2,…, t_M, and at most $4M+2$ constraints of the form $t_j - t_i + \alpha T + \beta T_1 \le C$, where $\alpha, \beta \in \{-1, 0, 1\}$ and C real.

A general linear program (LP) is known to be solvable in weakly polynomial time (Khachiyan, 1979). We will prove that the special-type LP problem (10)-(17) can be solved in *strongly* polynomial time. Cohen and Meggido (1994) gave an elegant strongly polynomial algorithm for solving the systems of linear inequalities of many variables with at most two variables per inequality. However, their method cannot be applied here since the problem (10)-(17) has up to four variables per inequality. (Of course, our algorithm is not valid for solving *any* LP with four variables per inequality; it worth mentioning that the general linear programming problem can be reduced in polynomial time to a LP with at most three variables per inequality).

The parametric algorithm for solving problem (10)-(17) consists of three stages. Its *first stage* is an extension of the Bellman-Ford algorithm for finding the longest path in graph $G = (N, A)$ with $|N| = M+1$ nodes and $|A| = 4M+2$ arcs, which is constructed like the graph G_1 in the previous section. The algorithm is based on the following fact from graph theory: finite critical paths to the nodes with numerical arc lengths exist if and only if there is no positive-length cycle in the graph. Each node has a multi-component label (potential); these potentials are re-calculated and improved from iteration to iteration. The potentials found after n-1 iterations have the form of max $(v_1,\ldots, v_L)$, where the terms $v_1,\ldots, v_L$, are constants or linear functions of T and T_1. They generate a set of inequalities (18) for variables T and T_1 for which there is no positive-length cycles in the graph (see (18) and Table 2 below).

Actually, at this stage, the algorithm converts the problem (10)-(17) with $2M+2$ variables into the problem of solving a system of $|A|$ non-linear inequalities (where $|A| \le 4M+2$) for two variables T and T_1. Each of these inequalities is of the following form (for the simplicity, the inequality index j here and below is omitted):

$$\max (v_1,\ldots, v_L) \le \max (u_1,\ldots, u_R), \tag{18}$$

where terms $u_1,\ldots, u_R$ are also constants or linear functions of T and T_1, whereas integer numbers L and R are defined by the algorithm.

Table 2. The parametric critical path (PCP) sub-algorithm

Step 1 . *// Initialization.* Enumerate all the nodes of G from 1 to $n = M+1$.
Assign labels $p^0(s) = p_1{}^0 = 0$, $p_j{}^0 = W(s \to j)$ if $j \neq s$;
Pred(s) $= \varnothing$ and $p^0(i) = -\infty$ to all other nodes i of G_1.

Step 2. // *Label correction.* For i:= 1 *to* n-1 do
For each arc $e = (t(e), h(e)) \in A_1$ compute

$$p^i(h(e)) := \max\{p^{i-1}(h(e)), p^{i-1}(t(e)) + W(e)\} \quad (19)$$

Step 3. // *Finding the family of disjunctive inequalities for T and* T_1.
For each arc $e = (t(e), h(e)) \in A_1$, write diwn the following system of functional inequalities

$$p^{n-1}(t(e)) + W(e) \le p^{n-1}(h(e)), \quad (20)$$

with respect to T and T_1. The inequalities (18) present the component-wise form of (20).

At the second stage each inequality (18) is reformulated as a union of R linear systems, $r = 1, \ldots, R$, each one containing $L+R$-1 inequalities as follows:

$$u_r \ge v_l,\ l=1,\ldots, L,\ u_r \ge u_q,\ q = 1,\ldots R,\ q \neq r. \quad (21)$$

In geometrical terms, each r-th linear system (21) defines a polygon denoted by BP_{jr} and called *basic*, where j is an arc number in A, $j = 1,\ldots, |A|$. All polygons BP_{jr} with the same index j are said to be *in layer j*. The family of all polygons lying in all $|A|$ layers is denoted by $\boldsymbol{F}$.

At the third stage, the algorithm uses the two-dimensional decomposition of all the polygons from the family $\boldsymbol{F}$ which is done by drawing horizontal and vertical lines through all the vertices of the basic polygons. Then it finds the most-left vertex in the T-T_1 plane that satisfies to at least one representative in each layer of the set $\boldsymbol{F}$.

Once the optimal values of T^* and T_1^* are found, the optimal values of all the t_k-variables (i.e., the optimal completion times of the processing operations) can be found as well, by replacing variables T and T_1 by their fixed values T^* and T_1^* in the expressions for the final node potentials or by solving the standard PERT/CPM problem on the graph with numerical data.

The complexity of the parametric procedure at the first stage is $O(n^3|A|)$. It is n^2 times larger than that for Bellman-Ford's algorithm for constant arc lengths because each label $p^i(h(e)) = \max(v_1,\ldots, v_L)$ is $O(n^2)$ long, where each v_i is $(C + bT + gT_1)$, C is a real number, and b and g are integers with the values in the range from -(n-1) to n-1.

The maximization operation in (19) in the PCP sub-procedure is carried out in a component-wise manner with the functional labels (which are of $O(n^2)$ components each). Computing each max-operator in (19) and the addition operation over the functional label in (19)-(20) require $O(n^2)$ elementary operations per each arc, totally Step 2 requires $O(n^3|A|)$ operations to be done for all arcs during n-1 iterations.

Further, finding the coordinates and boundary lines of the basic polygons in the family $\boldsymbol{F}$ defined by the system (18) at the second stage is a routine problem in computational geometry. In geometrical terms, at first, the algorithm finds all the basic

polygons BP_{jr} defined by (18). Given H half-planes, their intersection (that is, a corresponding basic polygon BP_{jr}) can be found in $O(H \log H)$. As $H = O(n^2)$ a single basic polygon can be found in $O(n^2 \log n)$ time. Any disjunctive inequality (18) defines a union of $O(n^2)$ basic polygons whereas the number of inequalities ("layers") in (18) is $|A|$. So the complexity of this stage is $|A|$ x $O(n^2)$ x $O(n^2 \log n) = O(n^5 \log n)$.

At the third stage, there are totally $O(n^5)$ vertices in all the polygons of $\boldsymbol{F}$. The first phase consists in building $O(n^{10})$ elementary polygons "covering" all the basic polygon BP_{jr} in $\boldsymbol{F}$. Each of the elementary polygons is either a rectangle, or a triangle, or a trapezoid. Their vertices form a lattice $\boldsymbol{L}$.in plane T-T_1. The elementary polygons are sorted from left-to-right and top-to-bottom, so totally the first phase is done in $O(n^{10} \log n)$. At the second phase the algorithm considers $|A|$ inequalities (21) at each of the elementary polygons. The intersection of $|A|$ different elementary polygons lying in $|A|$ layers but having the same coordinates of their vertices in lattice $\boldsymbol{L}$ produces a new *elementary intersection polygon* with at most $6|A|$ edges. Then the problem of minimizing T on each elementary intersection polygon is solved. Using the Megiddo (1983) or Dyer (1984) linear-time algorithm for solving 2-variable linear programs, the minimum T in all elementary intersection polygons is found in $O(|A|n^{10}) = O(n^{11})$ time. Finally, the vertex with minimum T is selected which is done in $O(n^{10})$. Thus, the algorithm complexity is $O(n^{11})$, or, in machine terms, $O(M^{11})$.

5 Concluding Remarks

Several modifications and extensions of the suggested algorithms can be derived. For example, the method in Section 4 can be easily modified to treat the precedence relations between operations described by an *arbitrary* acyclic graph, not only by a chain. This paper considers the problem with the *no-wait* condition traditionally occurring in the robotic flow lines, however the "*no no-wai*t" problem version can be solved by a slightly modified algorithm of the same complexity, $O(M^{11})$.

There are many unsolved cyclic scheduling problems occurring in practice, including multi-product, multi-stage, multi-robot and multi-gripper problems. Many of them are NP-hard. Analysis of their complexity and design of efficient algorithms are meritorious areas for future research. Another open question is the worst case complexity of the graph algorithms for the minimum (maximum) ratio problems. We conjecture that these algorithms can be enhanced to run in (weakly) polynomial time.

References

1. Brauner, N., Finke, G.: Optimal moves of the material handling system in a robotic cell. Int. J. Production Economics 74, 269–277 (2001)
2. Che, A., Chu, C., Levner, E.: A polynomial algorithm for 2-degree cyclic robot scheduling. European Journal of Operational Research 145(1), 31–44 (2003)
3. Chu, C.: A faster polynomial algorithm for 2-cyclic robotic scheduling. Journal of Scheduling 9(5), 453–468 (2006)
4. Cohen, E., Megiddo, N.: Improved algorithms for linear inequalities with two variables per inequality. SIAM Journal on Computing 23, 1313–1350 (1994)

5. Crama, Y., Kats, V., van de Klundert, J., Levner, E.: Cyclic scheduling in robotic flowshop. Annals of operations Research 96(1-4), 97–123 (2000)
6. Dantzig, G.B., Blattner, W., Rao, M.R.: Finding a cycle in a graph with minimum cost to time ratio with application to a ship routing problem. In: Rosenstiehl, P. (ed.) Theory of Graphs, Dunod, Paris, Gordon, Breach, New York, pp. 77–84 (1967)
7. Dawande, M.N., Geismer, H.N., Sethi, S.P., Sriskandarajah, C.: Througput Optimization in Robotic Cells. Springer, Heidelberg (2007)
8. Dyer, M.E.: Linear time algorithms for two- and three-variable linear programming. SIAM Journal on Computing 13(1), 31–45 (1984)
9. Hanen, C., Munier, A.: Cyclic scheduling on parallel processors: An overview. In: Chretienne, P., Coffman Jr., E.G., Lenstra, J.K., Liu, Z. (eds.) Scheduling Theory and Its Applications, pp. 194–226. Wiley, Chichester (1995)
10. Kampmeyer, T.: Cyclic Scheduling Problems. PhD thesis, University of Osnabrück, Germany (2006)
11. Kats, V., Lei, L., Levner, E.: Minimizing the cycle time of multiple-product processing networks with a fixed operation sequence and time-window constraints. European Journal of Operational Research 187(3), 1196–1211 (2008)
12. Kats, V., Levner, E.: Cyclic scheduling on a robotic production line. Journal of Scheduling 5, 23–41 (2002)
13. Kats, V., Levner, E.: A polynomial algorithm for 2-cyclic robotic scheduling: A non-Euclidean case. In: Gelbukh, A., Reyes-Garcia, C.A. (eds.) MICAI 2006. LNCS (LNAI), vol. 4293, pp. 439–449. Springer, Heidelberg (2006)
14. Kats, V., Levner, E., Meyzin, L.: Multiple-part cyclic hoist scheduling using a sieve method. IEEE Transactions on Robotics and Automation 15(4), 704–713 (1999)
15. Khachiyan, L.G.: A polynomial algorithm in linear programming. Sov. Math. Dokl. 20, 191–199 (1979)
16. Lee, T.E., Posner, M.E.: Performance measures and schedules in periodic job shops. Operations Research 45(1), 72–91 (1997)
17. Lei, L.: Determining the optimal starting times in a cyclic schedule with a given route. Computers and Operations Research 20, 807–816 (1993)
18. Levner, E., Kats, V.: A parametrical critical path problem and an application for cyclic scheduling. Discrete Applied Mathematics 87, 149–158 (1998)
19. Levner, E., Kats, V., Sriskandarajah, C.: A geometric algorithm for finding two-unit cyclic schedules in no-wait robotic flowshop. In: Levner, E. (ed.) Proceedings of the Workshop on Intelligent Scheduling of Robots and FMS, Holon, Israel, pp. 101–112. HAIT Press (1996)
20. Megiddo, N.: Linear time algorithms for linear programming in R3 and related problems. SIAM Journal on Computing 12(4), 759–776 (1983)
21. Romanovskii, I.V.: Optimization of stationary control of a discrete deterministic process. Kybernetika (Cybernetics) 3(2), 66–78 (1967)

Solving a School Timetabling Problem Using a Bee Algorithm

Carlos Lara, Juan J. Flores, and Félix Calderón

Universidad Michoacana de San Nicolás de Hidalgo
División de Estudios de Posgrado. Facultad de Ingeniería Eléctrica
Santiago Tapia 403 Centro. Morelia, Michoacán, México. CP 58000

Abstract. The timetabling problem consists in fixing a sequence of meetings between teachers and students in a given period of time, satisfying a set of different constraints. This paper shows the implementation of a Bee Algorithm (BA) to solve the Scholar Timetabling Problem. In the implemented BA, scout bees find feasible solutions while collector bees search in their neighborhood to find better solutions. While other algorithms evaluate every plausible assignment, the implemented algorithm only evaluates feasible solutions. This approach seems to be helpful to manage constrained problems. We propose a new measurement for replacing population that considers the evolutionary history of the bees as well as their fitness. Experimental results are presented for two real schools, where the algorithm shows promising results.

Keywords: Scholar Scheduling, Bee Algorithm, Decision Support System, Education.

1 Introduction

The Academic Timetabling Problem consists of scheduling a sequence of activities between teachers and students, satisfying a set of constraints. A large number of variants of the Academic timetabling problem have been studied in the literature, which differ from each other based on the type of institution involved. Schaerf et. al. [8] classifies them as: School Timetabling, Course Timetabling, and Examination Timetabling.

This paper focuses on the *School Timetabling Problem* (ST) also known as *Class–Teacher Timetabling Problem*. This problem consists of scheduling all the lectures of a week for a school, satisfying a set of constraints. Finding a good solution is sometimes difficult because this problem is computationally $\mathcal{NP}$–hard. The complexities and challenges of timetabling problems arise from the fact that there exists a large variety of constraints. In the timetabling literature, constraints are usually categorized into two types: hard constraints and soft constraints.

– **Hard Constraints.** These constraints cannot be violated under any circumstance. For instance, teachers cannot meet two classes at the same time.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 664–674, 2008.

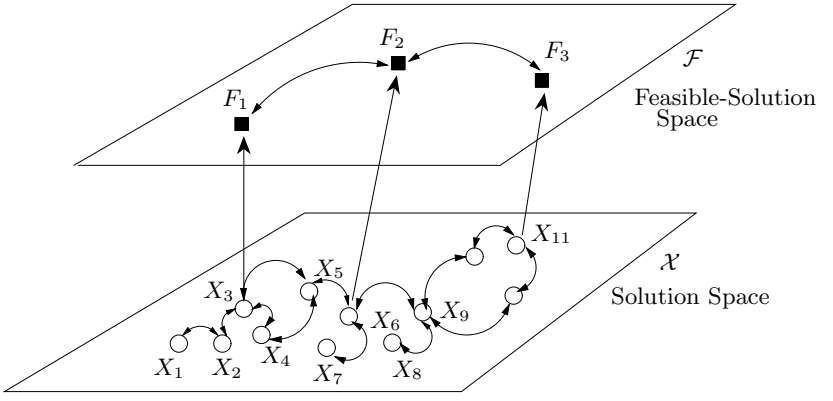


Fig. 1. Feasible–Solution Space

Analogously, a class cannot meet two lectures at the same time. A timetable assignment which satisfies all of the hard constraints is called a feasible solution.

– **Soft Constraints.** These constraints are desirable to solve but are not absolutely critical. In practice it is usually impossible to find feasible solutions that satisfy all of the soft constraints. The most common soft constraint in the school timetabling is to avoid empty time slots between class' activities.

Figure 1 illustrates that the search space $\mathcal{X}$ can be explored by using atomic moves. For example: X_2 is found by applying a single atomic move to X_1, X_3 is found from X_2, etc. Some instances (e.g. $X_3, X_6, \ldots$) do not violate hard constraints, and they are in the feasible–solution space $\mathcal{F}$. In ST as in many other combinatorial problems, we are looking for instances in the feasible–solution space. This paper proposes to use a Bee Algorithm (BA) to solve the ST problem, on which scout bees find feasible solutions while collector bees search in the neighborhood of these feasible solutions to find better solutions, keeping them feasible. The implemented algorithm iterates two basic stages:

1. **Exploration.** This stage performs a random search to find feasible solutions.
2. **Recollection.** This stage performs a local search in the neighborhood of a feasible solution to find another feasible solution. An example is shown in Figure 1: given F_1, the recollection stage tries to find another feasible solution– F_2. This stage finds another solution, which may or may not be better than the original one.

The rest of the paper is organized as follows: Section 2 describes the related work in this area. Problem statement is presented in Section 3. Then the representation used is introduced in Section 4. The implemented algorithm is presented in Section 5. Section 6 shows the results of computational tests that were conducted using real data. Finally, Section 7 presents our conclusions and proposes extensions to the work presented in this article.

2 Related Work

The timetabling problem is one of the scheduling problems that have been extensively studied. The solution approaches range from graph coloring to heuristic algorithms, including mathematical programming models and metaheuristics as well.

There are two different strategies used to explore the search space:

- those that search in the search space; in this strategy the unfeasible timetables are also included in the search space [9]. Unfeasibilities are usually highly penalized and the problem is seen as an optimization one, and
- those that search in the space of feasible solutions [2]; in this strategy unfeasible–solutions are not considered in the problem.

The implemented algorithm is one that searches in the space of feasible solutions. An example of this strategy is used in [2] to solve a slightly different problem –the University Course Timetabling Problem. Their approach consists of finding an initial population of feasible solutions, and then performing a local search by using a combination of neighborhood structures [1]. The reason for considering a combination of diverse neighborhood criteria is related to the diversification of search needed to escape from local optima [4]. We use a *Neighborhood Composition*, on which the neighborhood is composed by chains of atomic moves of variable length; although the chain length is limited to a maximum.

Our work is also related with memethic algorithms (MA); Memethic algorithms have provided very good results for a variety of scheduling problems [7]. MA are combinations of evolutionary based approaches with local search [7].

3 Problem Statement

The Scholar Timetabling problem is defined as follows. Given the following preliminary definitions:

- a set of teachers $T = \{t_i\}\ \ i = 1 \ldots n_T$,
- a set of classes $C = \{c_j\}\ \ j = 1 \ldots n_C$ (every class is a set of students having a common curriculum and studying together).
- a set of timeslots in a day $S = \{s_k\}\ \ k = 1 \ldots n_S$
- a matrix $R = [r_{i,j}]_{n_T \times n_C}$ called the *requirements matrix*, on which $r_{i,j} \geq 0$ is the quantity of lectures that teacher t_i must teach to class c_j in a week, and
- two matrices $Vt = [vt_{i,k}]_{n_T \times n_S}$ and $Vc = [vc_{j,k}]_{n_C \times n_S}$ that represent the availability of teachers and the availability of classes, respectively. Where $vt_{i,k}, vc_{j,k} \in \{0, 1\}$. For instance $vt_{i,k} = 0$ means that teacher i is unavailable at period k.

The problem is to assign meetings between teachers and classes in such a way that no teacher has two meetings at the same time and no class has two meetings

at the same time. More precisely, let us define $x_{i,j,k}$ to be 1 if teacher i meets class j at period k and 0 otherwise. Also, let us define the following functions, representing Soft Constraints:

- **Teacher Holes** $\mathcal{H}t(\cdot)$**.** Number of idle periods between lectures in the teachers assignment of a day.
- **Class Holes** $\mathcal{H}c(\cdot)$**.** Number of idle periods between lectures in the class assignment of a day.
- **Splits** $\mathcal{S}(\cdot)$**.** Number of periods that divide two or more lectures of the same class.

A mathematical formulation of the problem is the following [3]:

Determine $X = [x_{i,j,k}]_{n_T \times n_C \times n_S}$ such that it minimizes

$$f(X) = \alpha_1 \mathcal{H}t(X) + \alpha_2 \mathcal{H}c(X) + \alpha_3 \mathcal{S}(X) \tag{1}$$

Subject to

$$\forall i \in \{1 \dots n_T\}, \forall j \in \{1 \dots n_C\} \ \sum_{k=1}^{n_S} x_{i,j,k} = r_{i,j} \tag{2}$$

$$\forall i \in \{1 \dots n_T\}, \forall k \in \{1 \dots n_S\} \ \sum_{j=1}^{n_C} x_{i,j,k} \leq 1 \tag{3}$$

$$\forall j \in \{1 \dots n_C\}, \forall k \in \{1 \dots n_S\} \ \sum_{i=1}^{n_T} x_{i,j,k} \leq 1 \tag{4}$$

$$x_{ijk} \in \{0, 1\} \tag{5}$$

Where $\alpha_1 \dots \alpha_3$ are weight constants. Equation 2 constrains that for every teacher t_i and every class c_j, the sum of assigned timeslots must equal the requirement r_{ij}. Equation 3 constrains that teachers only have 0 or 1 activity associated to period k. Analogously, Equation 4 constrains that classes only have 0 or 1 activity associated to period k.

4 Representation of Constraints and Assignments

The previous section introduces matrices Vt an Vc, describing teacher's and classes' availability. In order to simplify the problem representation, this section introduces an equivalent representation for Vt an Vc using bit strings. This section also introduces how to represent a candidate solution X.

Let Ω be the set of bit strings $\mathbf{b} = \{0,1\}^{n_S}$ on which the ith bit represents whether timeslot i is assigned (1) or not (0). We can represent the teacher's availability matrix by $\mathbf{Vt} = \{\mathbf{vt_i} | i = 1, \dots, n_T\}$, class' availability matrix by $\mathbf{Vc} = \{\mathbf{vc_j} | j = 1, \dots, n_C\}$, and a candidate solution by the matrix $\mathbf{X} = [\mathbf{x}_{i,j}]_{n_T \times n_C}$; where $\mathbf{vt}, \mathbf{vc}, \mathbf{x}_{i,j} \in \Omega$. Given that each element $\mathbf{vt}_i \in \mathbf{Vt}$ is

a bit string, what was defined in the previous section as $vt_{i,k}$ is now the kth bit of $\mathbf{vt}_i$. For instance, a binary string $\mathbf{vt}_2 = 010000$ denotes that teacher 2 is only available at timeslot 2, $\mathbf{vc}_3 = 111000$ denotes that class 3 is available at timeslots 1, 2 and 3. Finally $\mathbf{x}_{2,3} = 0001100$ means that teacher 2 meets class 3 at slots 4 and 5.

Every assignment $\mathbf{x}_{i,j}$ must be compatible with availabilities of the ith teacher and the jth class. That is, $\mathbf{x}_{i,j} \vee \mathbf{vt}_i = \mathbf{vt}_i$, $\forall j \in \{1 \dots n_C\}$ and $\mathbf{x}_{i,j} \vee \mathbf{vc}_j = \mathbf{vc}_j$, $\forall i \in \{1 \dots n_T\}$.

Given a candidate solution $\mathbf{X}$, it is possible to retrieve the following information by using bitwise operations:

- **Used slots.** Used slots are slots on which teacher (or class) has one or more activities assigned. Teacher t_i's used slots are determined by

$$\mathbf{ut}_i = (\mathbf{x}_{i,1} \vee \dots \vee \mathbf{x}_{i,n_C}) = \bigvee_{j=1}^{n_C} \mathbf{x}_{i,j} \tag{6}$$

  and to find class j's used slots

$$\mathbf{uc}_j = (\mathbf{x}_{1,j} \vee \dots \vee \mathbf{x}_{n_T,j}) = \bigvee_{i=1}^{n_T} \mathbf{x}_{i,j} \tag{7}$$

- **Free timeslots.** Free timeslots of teacher i (or a class j), are slots on which teacher i (or class j) is available according to v_i (or vc_j) but has no activity assigned yet. Free slots are found by applying the XOR operation. For example, to find teacher i's free timeslots

$$\mathbf{dt}_i = \mathbf{vt}_i \oplus \mathbf{ut}_i \tag{8}$$

  and, to find class j's free slots

$$\mathbf{dc}_j = \mathbf{vc}_j \oplus \mathbf{uc}_j \tag{9}$$

  equations 8 and 9 hold because we enforce $\mathbf{ut}_i \vee \mathbf{vt}_i = \mathbf{vt}_i, \forall i$ and $\mathbf{uc}_j \vee \mathbf{vc}_j = \mathbf{vc}_j, \forall j$. That is, we restrict every assignment in the used slots to available slots (e.g. the case where $\mathbf{vt_i} = 0$ and $\mathbf{ut_i} = 1$ is not allowed).
- **Unfeasibilities.** There are two kinds of unfeasibilites:
  1. *Teacher–unfeasibility* occurs when a teacher has two assignments in the same period of time, and
  2. *Class–unfeasibility* occurs when a class has two or more assignments at the same time.

  To find unfeasibilities we simply count the number of bits for every slot in the current assignment of the teacher (or class). If the count for a determined slot is more than one then the slot has an unfeasibility. To reduce the complexity, the count is only performed on slots used by the teacher (or class).
- **Holes and Splits.** In order to determine holes and splits we need to introduce some concepts.

1. **The number of zeroes between ones.** The number of zeros between ones in a bit string $b \in \Omega$ is given by:

$$\mathcal{Z}(\mathbf{b}) = s_{end}(\mathbf{b}) - s_{start}(\mathbf{b}) - s_{used}(\mathbf{b}) + 1 \tag{10}$$

   where $s_{used}(\mathbf{b})$ is the number of slots used in the bit string b, $s_{start}(b)$ is position of the first 1 in $\mathbf{b}$, and $s_{end}(\mathbf{b})$ is the position of the last 1 in b. For instance, given $\mathbf{b} = 0101001000$, $s_{start}(\mathbf{b}) = 2$, $s_{end}(\mathbf{b}) = 7$, and $s_{used}(\mathbf{b}) = 3$; then the number of zeros between ones is $\mathcal{Z}(\mathbf{b}) = 7 - 2 - 3 + 1 = 3$. This formula counts the number of unused slots, whether they are contiguous or not.
2. **Clusters.** A cluster is a bit string $\mathbf{b}' \in \Omega$ which contains a single sequence of consecutive ones in b. Let us define $\Gamma(\mathbf{b}) = \{\mathbf{b}'_1 \dots \mathbf{b}'_\gamma\}$ as the set of all clusters of $\mathbf{b}$. For instance if $\mathbf{b} = 001110101$ then $\Gamma(\mathbf{b}) = \{00111000000, 000000100, 000000001\}$.
3. **Number of windows.** The number of windows of an assignment $y \in \Omega$ depends on the related availability $\mathbf{v}$, where $\mathbf{v}$ can be either $\mathbf{v_f}$ or $\mathbf{v_c}$. To evaluate the number of windows we use:

$$\mathcal{W}(\mathbf{y}, \mathbf{v}) = \sum_{\forall \mathbf{v}'_l \in \Gamma(\mathbf{v})} \mathcal{Z}(\mathbf{y} \wedge \mathbf{v}'_l) \tag{11}$$

   the number of windows of a bit string is relative to availability, i.e. there are no holes where there is no availability.

These definitions allow us to calculate:
the number of holes of the ith–teacher

$$\mathcal{H}t_i = \mathcal{W}(\mathbf{ut}_i, \mathbf{vt}_i) \tag{12}$$

the number of holes of the class c_j

$$\mathcal{H}c_j = \mathcal{W}(\mathbf{uc}_j, \mathbf{vc}_j) \tag{13}$$

and the number of splits of an assignment

$$\mathcal{S}_{i,j} = \mathcal{Z}(\mathbf{x}_{i,j}). \tag{14}$$

Finally, for a given assignment $\mathbf{X}$ the total number of teacher holes is $\mathcal{H}t(\mathbf{X}) = \sum_{i=1}^{n_t} \mathcal{H}t_i$, the number of class holes is $\mathcal{H}c(\mathbf{X}) = \sum_{j=1}^{n_c} \mathcal{H}c_j$ and the number of splits is $\mathcal{S}(\mathbf{X}) = \sum_{i=i}^{n_t} \sum_{j=1}^{n_c} \mathcal{S}_{i,j}$.

5 Bee Algorithm

The necessary and sufficient conditions for the existence of feasible solutions are discused in [5]. The following description assumes that feasible solutions exist.

The goal of the algorithm is to find a feasible and optimal solution. In other words, it must solve hard constraints (feasible solution) and satisfy as many

Algorithm 1. Bee Algorithm to solve STP

1: Initialize a population of m scout bees (algorithm 2).
2: **repeat**
3: Evaluate the fitness of the population by using Eq. 1.
4: Determine the size of the neighborhood (patch size).
5: Recruit bees for selected sites and collect.
6: In each site replace the scout bee for the local fittest bee
7: Create n new scout bees from random sites
8: Select the best m scouts according to Eq. 15
9: **until** converge

soft constraints as possible (optimal solution). We propose the use of the Bee Algorithm (BA) [6].

The BA mimics the food foraging behavior of swarms of honey bees. The BA performs a kind of neighborhood search combined with a random search. BA can be used for both combinatorial optimization and functional optimization. Its first step, initialization, consists on creating a population of n scout bees. Every scout bee finds a random feasible solution. In each iteration the BA evaluates the n sites using a fitness function, selects the best m sites, searches in the neighborhood of the m sites, and add a determined number of new random sites.

The algorithm we implemented is shown in Algorithm 1. To select the number of collector bees to each patch, we implemented a simple roulette wheel algorithm. The algorithm has a slightly but important difference with the basic algorithm, while the basic version uses only the fitness of the bees to keep m scouts, we suggest the use of a new measurement that considers the evolutionary history of the bees as well as their fitness:

$$g(\mathbf{X}) = f(\mathbf{X}) + \beta_1 e^{a'/\zeta_1} \tag{15}$$

where: the relative–age a' is the number of iterations since the fitness of the bee has improved. The new measurement allows the older bees to mature (develop a bette fitness), while allows newer bees to be competitive. If only a fitness–based evaluation is used, then newer scouts usually are not competitive against well explored neighborhoods. It implies that new scout bees are rarely selected for recollection. Parameters β_1, and ζ_1 must be tuned. If $\beta_1 = 0$ then $g(\mathbf{X}) = f(\mathbf{X})$ and the algorithm acts as the basic bee algorithm.

5.1 Searching Random Feasible Solutions: Scout Bees

Algorithm 2 shows how a random feasible solution is found by a scout bee. Firstly, it initializes a random solution $\mathbf{X}$ based on requirements R and subject to availabilities of teachers and classes, $\mathbf{Vt}$ and $\mathbf{Vc}$. After that, the infeasible solution $\mathbf{X}$ is repaired.

To get a feasible solution from the infeasible solution $\mathbf{X}$, Algorithm 3 iteratively performs two steps: (a) finds an infeasibility (teacher or class), and (b) replace it by a free slot. Every replacement is done randomly, therefore the found solution is also random.

Algorithm 2. Scout bees: searching a feasible solution from zero

Create a Random Matrix $\mathbf{X}$ *based on* R*:*
for all $\mathbf{r}_{i,j} \in R$ **do**
  $\mathbf{y}_{i,j} \leftarrow \mathbf{Vt}_i \wedge \mathbf{Vc}_j$
  $\mathbf{x}_{i,j} \leftarrow$ new random string with $\mathbf{r}_{i,j}$ ones, such that $\mathbf{x}_{i,j} \oplus \mathbf{y}_{i,j} = \mathbf{0}$
end for
Find a feasible solution by repairing $\mathbf{X}$ *(algorithm 3)*

Algorithm 3. Repairing an unfeasible solution

while $\mathbf{X}$ has unfeasibilities **do**
  Randomly select $\gamma \in \{0, 1\}$
  if $\gamma == 0$ **then**
    if there are teacher–unfeasibilities **then**
      Select a random teacher t_i that has unfeasibilities
      Select a random unfeasible slot of t_i and replace it for a free slot of t_i (computed with Eq. 8)
    end if
  else
    if there are class–unfeasibilities **then**
      Select a random class c_j that has unfeasibilities
      Select a random unfeasible slot of c_j and replace it for a free slot of c_j (computed with Eq. 9)
    end if
  end if
end while

5.2 Local Search: Collector Bees

While scout bees find feasible solutions from zero, collector bees do it by performing slight modifications to an initial feasible solution. Therefore collector bees perform a kind of neighborhood search. We implement three types of atomic moves for local search:

- **Moving a single lecture.** Move a single lecture to a different suitable slot, a suitable slot is a slot on which both the teacher and the class are available.
- **Moving a cluster of lectures.** The same case than the previous one, but move two or more consecutive lectures of a class-teacher pair to a different suitable slot.
- **Teacher swapping.** Given a class, select two teachers that teach to the class and swap their slots; both teacher slots must be available.
- **Class swapping.** Given a teacher, select two classes associated with the teacher and swap the two assigned slots; both class slots must be available.

Initially a collector bee follows the scout, that is create a copy $\mathbf{X}'$ of the scout assignment $\mathbf{X}$. After that, $\mathbf{X}'$ is modified by a single atomic move or a chain of atomic moves, these atomic moves can introduce unfeasibilities and the collector

Table 1. Experimental results for school A

	BA ($\beta_1 = 0$)	BA ($\beta_1 = 1$)	Manual
elapsed time	15 Min 24 Seg	15 Min 49 Seg	several days
best fitness	90	78	Unknown
total class holes	0	0	26
total teacher holes	30	26	64
total splits	0	0	13

bee needs to be fixed by using Algorithm 3. This reparation is tipically much easier than the one performed originally by scout bees.

6 Experimental Results

This section presents the experimental results. All code has been implemented in Java and runs on an Acer Aspire 3680–2125 with a 1.86 GHz processor. Two different schools A[1] and B [2] were used for the test, in those schools the timetabling problem takes several days of manual work. School–A managers are interested on keeping the same timetabling all days, while school–B managers are interested on an optimal timetable, and they allow different timetable for each day during the week. Both schools have enough room spaces because every class has an assigned room in their corresponding periods.

We have carried out experiments using the presented algorithm, the values for the parameters were: $\alpha_1 = 7$, $\alpha_2 = 3$, $\alpha_3 = 5$, $\beta_1 = 1$, and $\zeta_1 = 10$. The values of α_1, α_2 and α_3 reflect the scholar administrator preferences. The initial number of scout bees was $m = 40$, the maximum number of collector bees associated to each patch was 50.

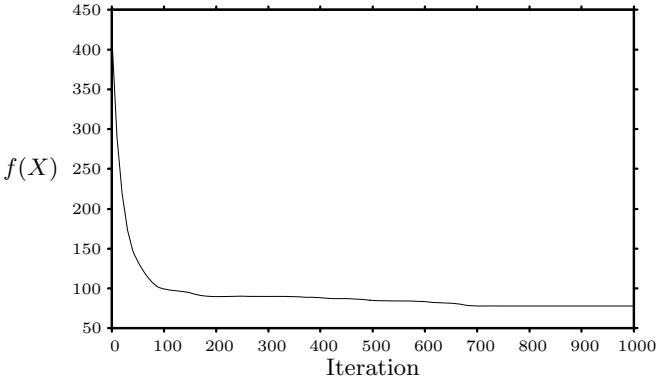
6.1 School A

The first timetabling data consists of 42 teachers, 23 classes, and 160 events per day. A typical evolutionary history for the best bee is shown in Figure 2. The number of iterations where 1000. We perform the experiment 100 times, the average results are shown in Table 1. Total class holes, total teacher holes and total splits are calculated for a day.

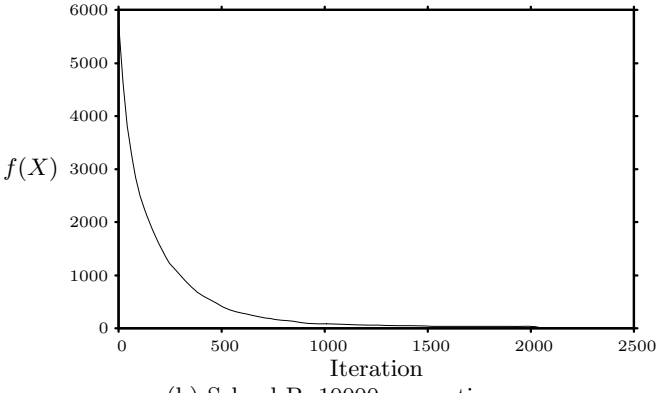
6.2 School B

The school B data consists of 86 teachers and 25 classes. School–B managers are interested on an optimal timetable, and they allow different timetable for each day during the week. To evaluate holes and splits we feed the algorithm with a list of day patterns. To evaluate $\mathcal{H}t(\cdot)$, $\mathcal{H}c(\cdot)$ and $\mathcal{S}(\cdot)$ we simply apply the

[1] South Guanajuato Institute of Technology.
[2] Michoacana University.



(a) School A, 1000 generations



(b) School B, 10000 generations

Fig. 2. Fitness of best bee for School A and B

bitwise–and operation over each assignment and day pattern. This makes sure that we do not consider the bit string representing a week as a long day. That kind of misinterpretation would change the semantics of the representation, thus yielding different measures for holes and splits.

A typical evolutionary history for the best bee is shown in Figure 2. The number of iterations where 10000. We perform the experiment 100 times, the average results are shown in Table 2.

Table 2. Experimental results for school B

	BA ($\beta_1 = 0$)	BA ($\beta_1 = 1$)	Manual
elapsed time	$\simeq$ 6 hrs	$\simeq$ 6 hrs	several days
best fitness	15	6	Unknown
total class holes at week	0	0	114
total teacher holes at week	5	2	43
total splits at week	0	0	0

7 Conclusions and Future Work

The Bee algorithm implemented to solve the scholar timetabling problem yields good results. The algorithm searches in the space of feasible solutions. In the implemented BA, scout bees find feasible solutions while collector bees search in their neighborhood to find better solutions.

The main contribution of this paper is to apply BA to STP by using a new measurement for replacing population that considers the evolutionary history of the bees as well as their fitness. This approach gives better results than using the plain fitness function.

Local search techniques give the possibility to start the search from any timetable, easily allowing for interactive construction and maintenance of timetables. In fact, once a timetable has been generated, it can be used as the starting point for a new search, after some constraints or the timetable have been manually modified. In the near future we want to test the technique presented for interactive construction of the timetable and to apply the bee algorithm in other contexts.

References

1. Abdullah, S., Burke, E., McCollum, B. (eds.): Using a Randomised Iterative Improvement Algorithm with Composite Neighbourhood Structures for the University Course Timetabling Problem. Computer Science Interfaces Book series. Springer Operations Research (2006)
2. Abdullah, S., Burke, E.K., McCollum, B.: A hybrid evolutionary approach to the university course timetabling problem. In: CEC (2007)
3. de Werra, D.: An introduction to timetabling. European Journal of Operational Research 19(2), 151–162 (1985)
4. Di Gaspero, L., Schaerf, A.: Multi-neighbourhood local search with application to course timetabling. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 263–278. Springer, Heidelberg (2003)
5. Neufeld, G.A., Tartar, J.: Graph coloring conditions for the existence of solutions to the timetable problem. Commun. ACM 17(8), 450–453 (1974)
6. Pham, D., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm a novel tool for complex optimisation problems. In: IPROMS 2006 (2006)
7. Rossi-Doria, O., Paechter, B.: A memetic algorithm for university course timetabling. In: Combinatorial Optimisation 2004 Book of Abstracts, Lancaster, UK, Lancaster University (2004)
8. Schaerf, A.: A survey of automated timetabling. In: Centrum voor Wiskunde en Informatica (CWI), vol. 115, page 33 (1995), ISSN 0169-118X
9. Schaerf, A.: Tabu search techniques for large high-school timetabling problems. In: Centrum voor Wiskunde en Informatica (CWI), vol. 88, page 17 (1996), ISSN 0169-118X

An Efficient Simulated Annealing Algorithm for Feasible Solutions of Course Timetabling

Juan Frausto-Solís[1], Federico Alonso-Pecina[1], and Jaime Mora-Vargas[2]

[1] Tecnológico de Monterrey Campus Cuernavaca,
69042 Cuernavaca, Morelos, México
[2] Tecnológico de Monterrey Campus Estado de México
{juan.frausto,a01125296,jmora}@itesm.mx

Abstract. Course Timetabling Problem (CTP) is a well known NP hard problem. Many classical randomized algorithms (as Genetic Algorithms, Simulated Annealing and Tabu Search) have been devised for this problem. For the previous PATAT benchmark, many of these old algorithms were able to find not only feasible solutions but even the optimal one. However, new harder CTP instances have recently proposed, which to obtain a feasible solution is a very hard challenge, and the previous algorithms do not perform well with these instances. Therefore, new algorithms for CTP should be devised. In this paper a new Simulating Annealing (SA) algorithm for CTP is presented. The algorithm shows a good performance not only with the old CTP instances but also with the new ones. This new SA implementation is able to find a feasible solution in instances where no other algorithm in the literature has been reported a success.

1 Introduction

Course Timetabling Problem (CTP) consists in fixing a sequence of meetings between teachers and students in a prefixed period of time (typically a week), satisfying a set of different classes of constraints [1]. The CTP instances proposed by PATAT have been intensively used as a benchmark for many researchers [2], [3]. This problem has two types of constraints: hard and soft. Hard constraints are those that rigorously must be satisfied in order to have a feasible solution. A hard constraint example is: "no student attends more than an event at the same time". Soft constraints are those that are desirable, but not completely indispensable; therefore a feasible solution is achieved when all the hard constraints are fulfilled, even though all the soft constraints are violated. In addition, for those CTP algorithms obtaining feasible solutions, their quality is better when more soft constraints are fulfilled. An example of a soft constraint is the following: "A student should not have more than two consecutive classes". Unfeasible solutions are completely neglected. Therefore, for CTP, any unfeasible solution fulfilling all the soft constraints is always worse than any feasible solution not fulfilling the complete soft constraints' set at all.

As is well known, some years ago, PATAT proposed a set of CTP instances where the objective function was defined as a weighted summation of hard and soft constraints [1]. For these instances, many algorithms based on classical meta heuristics

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 675–685, 2008.

have been proposed [1-6], [12-16]; among them Simulated Annealing [4], [11] and Tabu Search [13], [14] have reported the best results with these instances. However, sixty new hard instances have been proposed recently [7] which the big challenge is to obtain a feasible solution. For the previous PATAT benchmark, many of these old algorithms were able to find not only feasible solutions but even the optimal one. As a consequence, new algorithms are being developed all around the world to solve CTP. The challenge for finding a feasible solution is so big that for these instances, an optimal solution is not required and special algorithms are being specifically devised for them [8], [9]. However, in the literature, these new algorithms are unable to find a feasible solution in all these new instances.

The algorithm presented in this paper is based on Simulated Annealing (SA). As is well known, SA is a successful algorithm when their parameters are well tuned and a good neighborhood is used [18]. Kostuch developed an SA algorithm that won the CTP competition of PATAT with the old benchmark [11]. According to Kostuch, a very relevant element of his algorithm was its neighborhood, since it overcomes at least two other neighborhoods in SA, as was previously reported [1],[15]. Besides, in order to not violate hard constraints, extra timeslots are commonly open in successful approaches [3], [8], [9] to find a feasible solution in old PATAT's benchmark.

This paper presents a new SA algorithm based on a new neighborhood. This algorithm is able to find feasible solutions in instances where it has not been reported before. In addition, an implementation based on this algorithm was placed in seventh place in the last PATAT contest [19]. The paper is organized as follows: in section 2 the CTP problem is described. In section three, the instances to test the algorithms are presented. Section four shows the implementations and the results obtained with them are given in section five. Finally, in section six the conclusions and future work are discussed.

2 Problem Description

A feasible timetable is one in which all the events have been assigned in a specific timeslot, in a specific classroom (room in short), and all of the following hard constraints must be completely fulfilled:

i) H1: No student attends more than one event at the same time.
ii) H2: The room chosen for a particular event is big enough to house all the attending students and it satisfies all the technical features required by the event.
iii) H3: Only one event is hold in every room at any specific timeslot.

In addition, is desirable that the next soft constraints be satisfied:

iv) S1: A student should not have a class in the last slot of the day.
v) S2: A student should not have more than two classes consecutively.
vi) S3: A student should not have a single class on a day.

For CTP in PATAT contest, the next notation is used [10]: n Events: $E = \{e_1, e_2, .. , e_n\}$, m Students: $U = \{u_1, u_2, ..., u_m\}$, 45 Periods: $P = \{p_1, p_2, ..., p_{45}\}$, r Rooms: $A = \{a_1, a_2, ...,a_r\}$, r Roomsizes: $C = \{c_1, c_2, ... c_r\}$, and t Features: $F = \{f_1, f_2, ...,f_t\}$. There are also three binary matrixes [10]:

Matrix student/event: $D_{mxn;}$ (d_{il} = 1 if the student l attends the event i, and 0 otherwise).

$$D_{m\times n} = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots \\ d_{m1} & d_{m2} & \dots & d_{mn} \end{bmatrix}$$

Matrix room/feature: S_{rxt}, (s_{jf} = 1 if the room j satisfy the feature f, and 0 otherwise).

$$S_{r\times t} = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1t} \\ s_{21} & s_{22} & \dots & s_{2t} \\ \dots & \dots & \dots & \dots \\ s_{r1} & s_{r2} & \dots & s_{rt} \end{bmatrix}$$

Matrix event/feature: $Q_{nxt,}$ (q_{if} = 1 if the event i requires the feature f, and 0 otherwise).

$$Q_{n\times t} = \begin{bmatrix} q_{11} & q_{12} & \dots & q_{1t} \\ q_{21} & q_{22} & \dots & q_{2t} \\ \dots & \dots & \dots & \dots \\ q_{n1} & q_{n2} & \dots & q_{nt} \end{bmatrix}$$

To model hard and soft constraints, a three dimensional binary matrix $X_{n\times r\times P}$ is used, which the binary variable x_{ijk} is the representation of the event i, assigned to the room j in the period k (x_{ijk} =1 if this assignment is done and 0 otherwise). For instance, if x_{231} = 1, it means that the event 2 is held in the room 3, period 1. Therefore, the problem is to minimize the objective function Z (Number of soft constraints violated or *Scv*), subject to the hard constraints previously described.

2.1 Hard Constraints

In the next paragraphs the hard constraints are modeled.

H1 establishes that in any period k, at most one event $i \in Q_e$ can be programmed, where Q_e is the set of events which includes the event e and also all the events with a conflict with the event e. This constraint is modeled as the inequality (1), where i represents the event, j the room and k the period.

$$\sum_{i\in Q_e} x_{ijk} \le 1 \quad j = 1,\dots,r \quad k = 1,\dots,45 \quad e = 1,\dots,n \tag{1}$$

The constraint H2 can be divided in two: H2a: The room is big enough for all the attending students, and H2b: The room satisfies all the features required by the event. H2a can be also established as follows: For every room j where the event i is programmed, the capacity of the room j (i.e. c_j) will be higher or equal at the number of students that attend the event i. Therefore if a particular student is represented as l H2a can be written as:

$$b_i = \sum_{l=1}^{m} d_{li} \quad i = 1,\dots,n \tag{2}$$

Where d_{li} worths 1 if the student l attends the event i, 0 otherwise. As a consequence H2a can also be established as (3):

$$\forall x_{ijk} = 1, \quad b_i \leq c_j, \qquad i = 1,...,n \quad j = 1,...,r \quad k = 1,...,45 \tag{3}$$

H2b can be established as follows: For all the *k* periods, the room *j* must satisfy all the features required by every event *i* programmed in this room. Therefore:

$$x_{ijk} = 1 \Rightarrow \forall f \; q_{if} \leq s_{jf}, \qquad i = 1,..n \quad j = 1,...,r \quad k = 1,...,45 \tag{4}$$

Where q_{if} represents the feature *f* associated to the event *i*, and s_{jf} represents the feature *f* satisfied by the room *j*.

The constraint H3 establishes that in a period *k*, any room *j* can be held at most one event; then H3 can be written as:

$$\sum_{j=1}^{r} x_{ijk} \leq 1 \quad i = 1,...,n \quad k = 1,...,45 \tag{5}$$

2.2 Soft Constraints

Defining the set of the last period of every day as $V = \{k \mid k \bmod NUMPER = 0\}$, where $k = 1, 2, \ldots, 45$, and *NUMPER* is the number of periods in every day (9 for PATAT's case). In addition, the constraint S1 can be redefined as: No event *i* should be programmed in any period $k \in V$. So S1 can be written as:

$$\forall k \in V \quad \sum_{i=1}^{n} \sum_{j=1}^{r} x_{ijk} = 0 \tag{6}$$

S2 establishes that any student *l* should not have 3 or more events programmed in a row. Let be S_l the set of events containing the student *l*, S2 can be written as (7).

$$\forall \; v \in V \quad \sum_{k=v-NUMPER+1}^{v-2} \sum_{i \in S_l} \sum_{j=1}^{r} x_{ij(k+1)} x_{ij(k+2)} x_{ij(k+3)} = 0 \qquad l = 1,2,...,m \tag{7}$$

Finally S3 means that all the students should have programmed zero or more that one event per day (exactly one event is not permitted). Therefore:

$$\forall v \in V \quad \sum_{k=v-NUMPER+1}^{v-2} \sum_{i \in S_l} \sum_{j=1}^{r} x_{ijk} \neq 1 \qquad l = 1,...,m \tag{8}$$

3 Problem Instances

Two benchmarks are used in this paper; in the former, an instances' generator is used: every instance has at least one perfect solution (i.e. zero violated constraints). This benchmark was used for the PATAT competition [10], every instance Competitionx is called *Cx*, (e.g., Competition01 is C01, and so on). In the table 1, the features of PATAT's instances are presented. For the second benchmark [8], every instance has at least one feasible solution (i.e. zero hard constraints violated); in table 2, the features of these instances are presented. For this benchmark, there are sixty instances, which twenty of them are small (*S01*, *S02*, ... *S20)*, twenty medium (*M01*, *M02,...M20)* and twenty big (*B01, B02, B03,...B20*).

Table 1. PATAT´s Benchmark

Instance	Events	Rooms	Features	Students
C01…C20	350-440	10-11	5-10	200-350

Table 2. Classification of the second benchmark

Instance	Events	Rooms	Features	Students
Small	200-225	5-6	3-10	200-1000
Medium	390-425	10-11	8-10	400-1000
Big	1000-1075	25-28	10-25	800-1200

4 Algorithm Description

As we mentioned before, Simulated Annealing (SA) is one of the most successful metaheuristic for CTP, for instance for the contest of PATAT in 2002 [10], a SA algorithm was the winner [11]. In the present paper a SA is presented and it achieves again very good results. This SA, considers: the search space, the neighborhood' criterion, the objective function, the initial and final temperature T_0 and T_f, respectively, and the cooling scheme temperature. The last one is modeled by a geometrical cooling function $T_k = \alpha T_{k-1}$, where k is the iteration number, and α is a parameter close to 1 (0.99 in this paper). To obtain a feasible solution, each event is scheduled in a room satisfying its requirements, and avoiding every student has more than one event scheduled in the same timeslot [8], [9]. In order to not violate any hard constraint, extra timeslots are added, and a feasible solution is searched using each time a certain amount of timeslots in the assignment; the search's process is stopped when the extra timeslots are zero. During this search, the distance of any solution to the feasibility region is measured as the minimum number of events required to be rescheduled in order to obtain a feasible solution. The initial solution is obtained by setting all the events to a suitable room, and avoiding causing any violation of hard constraints, even though extra timeslots have to be added.

A key issue to find good solutions with SA is the cooling scheme (initial and final temperatures, and the function for reducing the temperature). The algorithm has three phases, in the first one the length Markov chain (iteration's number of the k metropolis cycle or L_k) grows slowly (from T_0 to T_b) in the second one, L_k grows with the analytical tuning method [17] (from T_b to T_t), and in the third phase the markov chain remains constant (from T_t to T_f).

The initial temperature T_0 of the algorithm is the temperature corresponding to the worst solution, $T_0 = 470n + e$, where n is the number of students and e is the number of events of the problem. This formula is obtained for the PATAT's parameters (45 periods, number of students, and number of events). The second temperature, named T_b, corresponds to the maximum deterioration of the objective function [17], and it is repeated in (9) for completeness. In this formula, the maximum possible deterioration, with the new neighborhood is: K*(maximum number of students per event).

$$T_0 = \frac{-\Delta Z_{V\max}}{\ln\left(P^A\left(\Delta Z_{V\max}\right)\right)} \tag{9}$$

From T_0 to T_b a quenching annealing process is done; it means a SA process where the temperature is quickly decreased until T_b. The temperature from T_0 to T_b is decreased more quickly not because α is different but because the number of iterations into the metropolis cycle is reduced. Therefore, the first SA process (i.e. the quenching process) finishes at T_b. From T_b to T_t an analytical SA is executed (details are explained later). T_t represents the temperature where the deteriorations of the objective function started to be least abrupt and was measured experimentally. Because the Metropolis cycle reaches the stochastic equilibrium as a markovian process, T_t can be used to determine the number of Markov chain in this cycle [17]. T_f represents the stop criterion of a classical SA. Both T_t and T_f are obtained experimentally as 1000 and 0.01, respectively. In every iteration, the temperature is decreased geometrically as $T_n = \alpha T_{n-1}$, with $\alpha = 0.99$. For CTP, SA obtains a new solution by using a certain neighborhood criterion; currently simple and double neighborhoods are used**.** In a simple neighborhood a new solution is obtained by changing an event from a period or room to another one. In the second neighborhood two events are taken and their respective rooms and periods are interchanged. Nevertheless, both of these neighborhoods are very inefficient for the new CTP instances. In this paper a new neighborhood was designed; the general idea is to promote feasibility by increasing the number of movements (NumMove) which do not violate any hard constraint. This neighborhood tries to bring closer the new solution to the feasible region every time it is applied. This new neighborhood is the next one:

Step 1: Select a random event,
Step 2: Calculate the feasible neighborhood (Only the movements not violating hard constraints are considered). If NumMove = 0, go to step 1.
Step3: If NumMove>=1, then do a random selection among the movements and make the chosen movement.

The Markov chains length L of the metropolis loop (the inner one) are also tuned dynamically. In [17], an analytical method to determine the longitude L_i for i iteration is presented. In this method, L_i is determined establishing the relationship between the cooling function and the Markov chain length. In the first Metropolis cycle the L parameter is equal to one and is increased according to a β-parameter, until, in the last cycle of Metropolis, it reaches a maximum value L_{max}, (10,000 in this case):

$$L_{max} = \beta^n L_1 \tag{10}$$

Where

$$n = \frac{\ln T_b - \ln T_t}{\ln \alpha} \tag{11}$$

$$\beta = \exp \frac{\ln L_{\max} - \ln L_1}{n} \tag{12}$$

From T_t to T_f a standard SA is executed, with a constant length of Markov chain.

The pseudocode of the algorithm implemented is:

```
Begin
   x = initial_solution
   num_t =  get_number_of_timeslots(x);
   BestCost = costIni = f(x)  // f(x)= objective function
   T = 470 * num_students + num_events
   Tb = - 8 * MAX_CAPACITY_ROOM / log (0.95);
   Tf = 0.01;  Tt = 1000;      L = 1;  Lmax = 10000;
   α = 0.99;  Iter = 0; n = (ln(T) - ln(Tt))/Ln(α);
   Beta = Exp((ln(Lmax)-ln(L))/n);
   While (T > Tf)
      While (iter < L)
         x_new = perturb(x);
         costNew = f(x_new);  costDif = costNew - costIni;
         t = get_number_of_timeslots(x_new);
         if (costNew <= 0) or t < num_t then
            num_t = t; costIni = costNew; x = x_new;
         else r = rand()
              if (r < exp(-costDif/T)) then
               costIni = costNew;   x= x_new;
              End_if
         End_if
         if (BestCost > costIni) then
              x* = x;   BestCost = costIni;
         End_if
         iter = iter + 1
       End_While
       T = T * α;
       if(T > Tb) then L = L +1;
       else if(L < Lmax)then L = Beta * L;
   End_While
End
```

5 Results

As we mentioned before, for CTP algorithms the goal is to find not only feasible solutions but also the optimal one with the minimum execution time. In addition, for the new (and very hard) instances, a very big challenge for the algorithms is to obtain a feasible solution with the smallest execution time. Therefore, the quality solution of a result is defined as the distance to the feasibility region which is measured as the minimum number of events required to be rescheduled in order to obtain a feasible solution. Nevertheless, there is an important issue related to quality solution in CTP community: a feasible solution Xi is always better than an unfeasible solution Xj, no matter the number of soft constraints violated in these solutions. For instance, let us suppose the feasible solution Xi violates all its soft constraints while the unfeasible Xj does not fulfill a single hard constraint hj, which does not violate any soft constraint; even in this case Xi will be better than Xj.

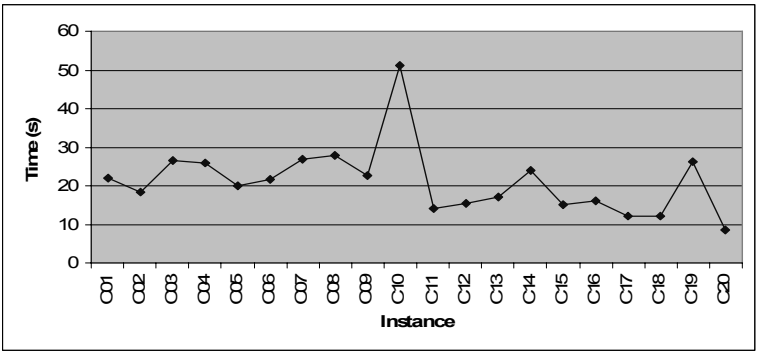


Fig. 1. Time results for the first benchmark

The average results are determined in this paper for comparison proposals with other algorithms; therefore the average of the distances to the feasible region was calculated. For each instance of the two benchmarks the algorithm were run 20 times. For the first benchmark, a feasible solution was obtained in all the executions, and the average execution time obtained is shown on figure 1. In figure 2, the number of soft constraints violated for these instances are presented, showing an acceptable performance.

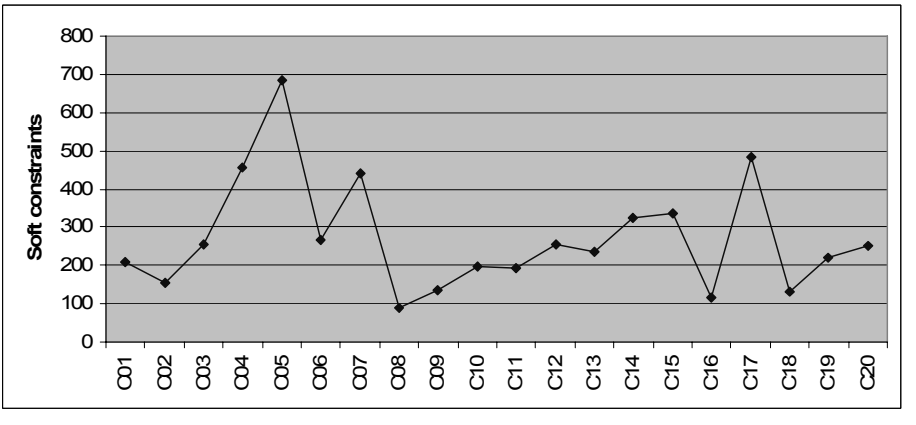


Fig. 2. Average of soft constrains violated for the first benchmark

The next figures show the quality results of the algorithm in the second benchmark. In these tables two number are provided, the first one is the average of 20 runs and the second one is the best result of 20 runs (the latter is parenthesized). The results obtained by SA are compared with the results of [8] and [9], and they are available in [7]. The Genetic algorithm of [8] is labeled as GA01, the genetic algorithm of [9] is labeled as GA02 and the Heuristic algorithm of [9] is labeled as H01, the implementation of SA of this article is labeled as SA.

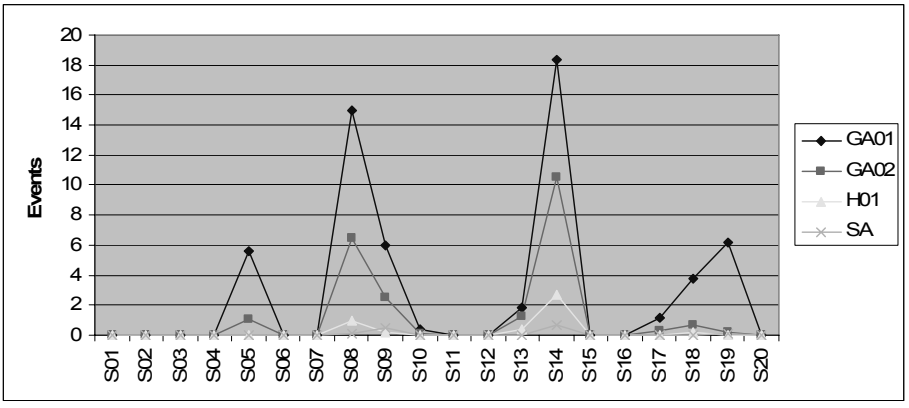


Fig. 3. Results for small instances. The graphic shows the average of the distance to feasibility by the algorithms. The best results are achieved by SA.

SA was able to find a feasible solution for all the small instances; notice than also H01 was able to achieve that. But, only the SA proposed in this paper is able to find at least a feasible solution for all the medium instances; SA finds feasible solutions in instances which no other algorithm was reported as able to find a feasible solution: these instances are M07, M09 and M16. Figure 4 shows the results in the medium instances. According to the results shown in figure 2, SA is the algorithm obtaining the best results for the big instances; it finds feasible solutions in the instances B04, B05, B08, B09, B10, B11 and B16, where again no other algorithm has reported a success. In the instances where the proposed algorithm did not find a feasible solution, their average of events without an assignment is smaller than that of the other algorithms. Significance test of these results clearly show that the proposed algorithm outperform the classical SA, GA and the Heuristic proposed by Lewis and Patcher [9]. Due to lack of space, the graphics showing this test are omitted in this paper but it was done according to [15].

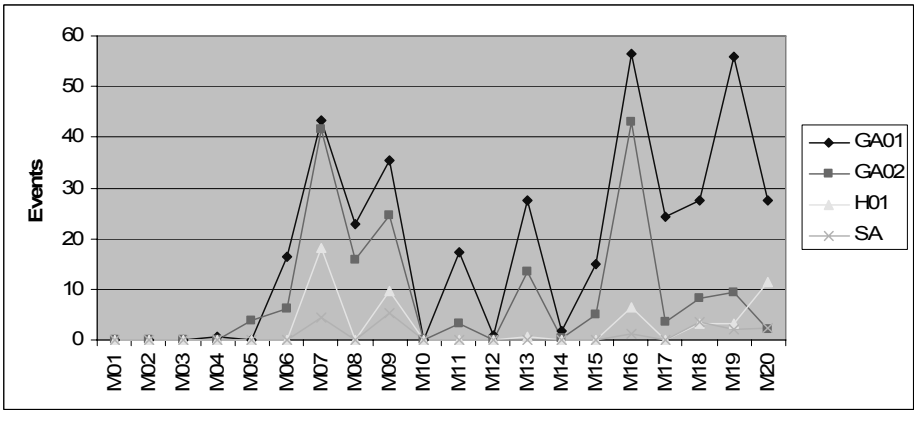


Fig. 4. Results for medium instances. The graphic shows the average of the distance to feasibility by the algorithms. The best results are achieved by SA.

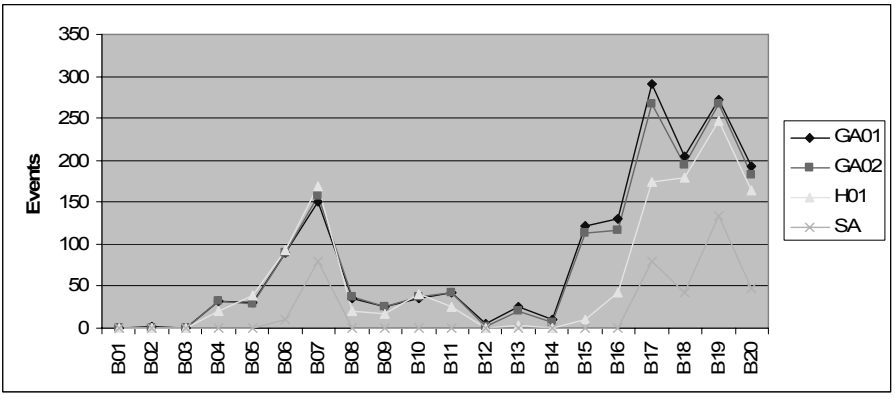


Fig. 5. Results for big instances. The graphic shows the average of the distance to feasibility obtained by the algorithms. The best results are achieved by SA.

6 Conclusions

In this paper, an implementation of Simulated Annealing to obtain feasible solutions in the problem of Course Timetabling is presented. The implementation obtains feasible solutions in all the instances of the first benchmark. In the second benchmark , SA obtains feasible solutions in all instances type small. The algorithm obtains feasible solutions in all instances type medium, inclusive in instances M07, M09, and M16, where no feasible solution has been reported before. In addition, the algorithm proposed here is also the most effective in the instances type big, because: a) SA has the minimum average of events without an assignment and b) SA has been able to find a feasible solution in the seven big instances B04, B05, B08, B09, B10, B11 and B16, which no other algorithm has reported a success. According to these results, SA is the most robust and effective implementation to obtain feasible solutions than those reported up to now in the literature. A future work is to hybridize this algorithm with other ones. Particularly, we obtained good results with the hybridization of SA with Tabu Search in the last PATAT conference. Therefore, it would be interesting to develop a similar hybrid algorithm, but using the SA algorithm presented in this paper.

References

1. Rossi-Doria, O., Sampels, M., Biratrari, M., Chiarandini, M., Dorigo, M., Gambardella, L.M., Knowles, J., Manfrin, M., Mastrolilli, L., Paetcher, B., Paquete, L., Stützle, T.: A comparison of the performance of different metaheuristic on the timetabling problem. Napier University, Université Libre de Bruxelles, Technische Universitaet Darmstadt (2002)
2. Socha, K., Knowles, J., Sampels, M.: A MAX-MIN Ant System for the University Timetabling Problem. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) Ant Algorithms 2002. LNCS, vol. 2463, pp. 1–13. Springer, Heidelberg (2002)

3. Krzysztof, S.: MAX-MIN Ant Systems for International Timetabling Competition. International Timetabling Competition, March 31 (2003)
4. Bykov, Y.: The Description of the Algorithm for International Timetabling Competition. Timetabling Competition, University of Nottingham, School of Computer Science & IT, Wollaton Road, Nottingham NG8 1BB, UK
5. Halvard Arntzen, Arne Lokketangen. A local search heuristic for a university timetabling problem, March 31 (2003)
6. Frausto-Solis, J., Alonso-Pecina, F., Larre, M., Gonzalez-Segura., C., Gomez-Ramos, J.L.: Solving the Timetabling Problem with three heuristics. WSEAS Transactions on Computers 5(11), 2849–2855 (2006)
7. Lewis, R.: (June 2007), http://www.emergentcomputing.org/timetabling/harderinstances
8. Lewis, R., Paechter, B.: Application of the Grouping Genetic Algorithm to University Course Timetabling. In: Raidl, G.R., Gottlieb, J. (eds.) EvoCOP 2005. LNCS, vol. 3448. Springer, Heidelberg (2005)
9. Lewis, R., Paechter, B.: Finding Feasible Timetables using Group-Based Operators. Technical Report NUS-2005-01, Centre for Emergent Computing, Edinburgh, Scotland 2005. IEEE Transactions on Evolutionary Computation (2005) (accepted for publication)
10. International Timetabling Competition, Consultant date: Wednesday (June 2007), `http://www.idsia.ch/Files/ttcomp2002/`
11. Kostuch, P.A.: The University Course Timetabling Problem with 3-Phase approach. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 251–266. Springer, Heidelberg (2005)
12. Di Gaspero, L., Schaerf, A.: Timetabling Competition TTComp 2002: Solver Description, March 31 (2003)
13. Cordeau, J.-F., Jaumard, B., Morales, R.: Efficient Timetabling Solution with Tabu Search, March 31 (2003)
14. Di Gaspero, L., Schaerf, A.: Timetabling Competition TTComp 2002: Solver Description, March 31 (2003)
15. Caballero, S.A.: Desarrollo y aplicación de un algoritmo híbrido de cómputo evolutivo y Recocido Simulado para la solución de problemas de Timetabling (The development and application of a hybrid algorithm of Evolutionary Computation and Simulated Annealing for the solution of problems of Timetabling) Master Thesis, ITESM Campus Estado de México (July 2007)
16. Frausto-Solis, Federico, A.-P.: Analytically Tuned Parameters of Simulated Annealing for the Timetabling Problem. In: WSEAS International Conferences Puerto De La Cruz, Tenerife, Canary Islands, Spain, December 14-16 (2007)
17. Sanvicente-Sanchez, H., Frausto Solís, J.: Method to Establish the Cooling Scheme in Simulated Annealing Like Algorithms, México (2003)
18. Aarts, E., Korst, J.: Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing. hichester [England]. Wiley, New York (1989)
19. Frausto-Solis, J., Alonso-Pecina, F.: FRALONSO: A Hybrid Simulated Annealing-Tabu Search Algorithm for Post Enrolment Course Timetabling. In: PATAT Conferences 2008, Montreal Canada (accepted, 2008)

On the Definition of Essential and Contingent Properties of Subjective Belief Bases

Ebrahim Bagheri and Ali A. Ghorbani

Faculty of Computer Science,
University of New Brunswick, Fredericton, Canada
{e.bagheri,ghorbani}@unb.ca

Abstract. In this paper, we introduce several features of subjective belief bases from both individualistic and collective perspectives and hence provide suitable essential and contingent properties for such belief bases. Essential properties reflect the attributes of a belief base being considered in vacuum, whereas contingent properties of a belief base reveal its characteristics with regards to the rest of its peer belief bases. Subjective belief bases employ values from Subjective logic, a type of probabilistic logic that explicitly takes uncertainty and belief ownership into account, to represent the priority information of the formula in each belief base. We show that subjective belief bases are a generalization of prioritized belief bases whose formula are annotated with their degree of necessity from possibilistic logic. We also discuss the role of essential and contingent properties in defining suitable belief base ordering functions.

1 Introduction

In a belief revision or belief merging process, there is generally a set of not necessarily consistent belief bases that are either employed to update the current state of belief or to develop a single representative belief base conveying consistent information about the universe of discourse. These processes are easily performed when the belief bases are consistent or non-overlapping. In cases where the belief bases contain mutually discrepant content, the decision about the preferred propositions requires prioritization between the conflicting beliefs. A straightforward solution is to trust the most reliable information source [19]. In such a strategy, reliability has to be either acquired through the observation of the credibility of the information source or through third-party accreditation. An alternative approach is to take advantage of the belief bases content to reason about the plausibility of their arguments. Various measures and properties have been introduced in the related literature that allow the comparison of belief bases.

The Dalal (Hamming) distance is a fundamental metric, which measures the quantity of conflict between two belief base interpretations by calculating the number of conflicting atoms [3]. Hunter builds on this metric by developing the degree of conflict between two belief bases, which is employed for measuring arguments' believability [7]. Here, the degree of conflict of two belief bases is

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 686–698, 2008.

defined as their Dalal distance divided by the maximum possible distance between any pair of models in their union. This measure of conflict only considers the normalized quantity of conflict as the criterion for the distance of two belief bases. More recently, Qi et al. [18] have argued that the distance between two belief bases is not only dependent on the quantity of their conflict, but also on their capacity to agree. In their definition, the capacity to agree is measured by the amount of explicit agreement between the bases (strong agreement) and the quantity of undisputed agreement (weak agreement). Konieczny et al. have also developed a model for defining the relationship between several pure propositional belief bases, which is based on the degree of contradiction and ignorance of the belief bases [11].

Some researchers have lately focused on the manipulation of inconsistent belief bases in the presence of priority information. In these belief bases, propositional formula are annotated with priority information mainly in the form of probability values, possibility measures, or belief-theoretic mass assignments [17]. Here, the priority information can be employed to develop more complex measures for comparison. For instance, the degree of *non-specificity* is a measure of the degree of uncertainty of a possibility distribution attributed to a belief base [6]. Based on this measure, the degree of *information closeness* of two belief bases can be computed [5]. Information closeness can behave as a distance measure. In both possibilistic and probabilistic belief bases, the measure of *information divergence* can also be used to assess the distance between two belief bases [12]. Jenhani et al. have proposed a composite measure of similarity of two possibility distributions called *information affinity* [8]. This measure is based on both the Manhattan distance of the two distributions and their degree of inconsistency. Along the same lines, and for a belief-theoretic setting, Liu defines a two-dimensional measure of conflict consisting of the maximum extent of the difference between the betting commitments to all the possible worlds, and the belief mass assigned to the empty set after combination by the Dempster's rule of combination [14].

In this paper, we introduce some interesting properties of belief bases annotated with priority information. These properties fall into two categories, namely: *essential* and *contingent* properties. Essential properties reflect the attributes of a belief base from an individualistic viewpoint, whereas contingent properties of a belief base reveal its collective characteristics with regards to the rest of the peer belief bases. We propose the employment of an extension of Dempster-Shafer theory known as Subjective logic for augmenting propositional belief bases with priority information, and show how it generalizes prioritized belief bases that are annotated with possibilistic information.

This paper is organized as follows. Section 2 provides some preliminaries, then the essential properties of belief bases are introduced. Subjective prime implicants are defined in Section 4, followed by the introduction of contingent properties in Section 5. A discussion on ordering belief bases is given in Section 6, and the paper is then concluded.

2 Preliminaries

Classical Logic: Throughout this paper, we let $\mathcal{L}$ be a propositional language over a finite alphabet $\mathcal{P}$ of propositional symbols. Ω represents the set of possible interpretations. An interpretation is a function from $\mathcal{P}$ to $\{\top, \bot\}$, where $\top$, and $\bot$ denote truth and falsehood, respectively. An interpretation ω is a model of formula ϕ, noted as $\omega \models \phi$ which makes the formula true. Furthermore, let ϕ be a formula, $mod(\phi)$ is employed to denote the set of models of ϕ, i.e. $mod(\phi) = \{\omega \in \Omega \mid \omega \models \phi\}$. Classical propositional logic deduction is represented using $\vdash$. Two formula such as ϕ and φ are equivalent, expressed $\phi \equiv \varphi$, if and only if $\phi \vdash \varphi$ and $\varphi \vdash \phi$. A formula ϕ satisfying $mod(\phi) \neq \bot$ is considered to be consistent. A clause C is a disjunction of literals.

Evidence Theory: Dempster-Shafer (DS) theory is basically an extension to probability theory where probabilities are assigned to sets as opposed to singleton elements [20]. The employment of the DS theory requires the definition of the set of all possible states in a given setting, referred to as the frame of discernment represented by θ. The powerset of θ, denoted 2^θ, incorporates all possible unions of the sets in θ that can receive belief mass. The truthful subsets of the powerset can receive a degree of belief mass; therefore, the belief mass assigned to an atomic set such as $\psi \in 2^\theta$ is taken as the belief that the given set is true. The belief in ψ is interpreted as the absolute faith in the truthfulness of ψ, which not only relies on the belief mass assigned to ψ, but also to belief masses assigned to subsets of ψ. DS theory provides two important functions: Belief (Bel) and Plausibility (Pl) functions. The belief and plausibility functions are interrelated i.e. $Pl(\psi) = 1 - Bel(\psi')$.

Subjective logic adds to DS theory by introducing two extra functions: Disbelief and Uncertainty [9]. A belief expression in Subjective logic is therefore defined as a 3-tuple $\chi^A_\psi = (b^A_\psi, d^A_\psi, u^A_\psi)$ also known as the opinion of expert A about hypothesis ψ.

Definition 1. *Let $\chi_\psi = (b_\psi, d_\psi, u_\psi)$ be a subjective opinion. The elements of χ_ψ can be calculated from a DS basic belief mass assignment by allowing:*

$$b_\psi = Bel(\psi). \tag{1}$$

$$d_\psi = \sum_{\varphi \cap \psi = \varnothing} m_\theta(\varphi), \quad \varphi, \psi \in 2^\theta. \tag{2}$$

$$u_\psi = \sum_{\varphi \cap \psi \neq \varnothing, \varphi \not\subseteq \psi} m_\theta(\varphi), \quad \varphi, \psi \in 2^\theta. \tag{3}$$

A Subjective Belief Base (SBB) is a set of propositional formulae annotated with subjective opinions in the form of $B = \{(\phi^*_i, \chi_{\phi_i}) : i = 1, .., n\}$, where χ_{ϕ_i} is a subjective opinion such that $\chi_{\phi_i} = (b_{\phi_i}, d_{\phi_i}, d_{\phi_i})$. The classical propositional form of B is represented by B^* $(\{\phi_i | (\phi^*_i, \chi_{\phi_i}) \in B\})$.

Proposition 2. *[15] For any given possibility distribution such as π a basic belief mass assignment function within the context of DS theory can be always derived such that the corresponding belief function is consonant.*

This proposition shows that necessity and plausibility measures in possibility theory are associated with belief and plausibility functions in DS theory; therefore, a corresponding consonant belief function can always be derived to represent any given possibility distribution.

Proposition 3. *Let $\mathcal{A} = \{(\phi_1, a_1)..., (\phi_n, a_n)\}$ be a prioritized belief base such that the propositional formula (ϕ_i) are augmented with their degree of necessity (a_i) in possibilistic logic. A corresponding subjective belief base (image of $\mathcal{A}$), denoted $\mathcal{A}_s$, can be derived to represent $\mathcal{A}$.*

Proposition 3 is an important result of Definition 1 and Proposition 2, and shows that any prioritized belief base can be converted into a subjective belief base. The consequence of this proposition is that subjective belief bases are a generalization of prioritized belief bases and any of the properties defined for a subjective belief base is applicable to a prioritized belief base through its image in subjective form.

3 Essential Properties

In this section, we discuss some of the basic essential properties of subjective belief bases that are apparent from a belief base's individualistic attributes without attention to the content of other peer belief bases. First, we introduce these features on single formulae of the belief bases and later extend them to accommodate belief bases.

Definition 4. *Let B be a subjective belief base, and $(\phi, \chi) \in B$ be a formula in B where $\chi = (b_\chi, d_\chi, u_\chi)$. The degree of ambiguity of (ϕ, χ), denoted $\mathcal{AMB}(\phi, \chi)$, is defined as:*

$$\mathcal{AMB}(\phi, \chi) = -\left(\frac{1}{e^{1-(b_\chi + d_\chi)}} - 1\right) \times \epsilon. \tag{4}$$

A normalization factor, ϵ, fits $\mathcal{AMB}(\phi, \chi)$ to $[0, 1]$.

Ambiguity provides the basis to calculate the degree of confusion of the information source with regards to the exact fraction of belief that should be assigned to a given formulae.

Definition 5. *Let B be a subjective belief base, and $(\phi, \chi) \in B$ be a formula in B where $\chi = (b_\chi, d_\chi, u_\chi)$. The degree of indecisiveness of (ϕ, χ), denoted $\mathcal{IND}(\phi, \chi)$, is defined as:*

$$\mathcal{IND}(\phi, \chi) = \left(\frac{1}{e^{|b_\chi - d_\chi|}} - \frac{1}{e}\right) \times \epsilon'. \tag{5}$$

A normalization factor, ϵ', fits $\mathcal{IND}(\phi, \chi)$ to $[0, 1]$.

Indecisiveness is a measure of the ability of the information source to firmly state a given formula. The further away the degrees of belief and disbelief for a given formula are, the stronger and more decisive the formula is.

Proposition 6. *Let B be a subjective belief base, $(\phi, \chi) \in B$ be a formula in B where $\chi = (b_\chi, d_\chi, u_\chi)$, and $\mathcal{AMB}(\phi, \chi)$ and $\mathcal{INC}(\phi, \chi)$ be the degrees of ambiguity and indecisiveness of (ϕ, χ), respectively. The following two relations hold:*

- *if $\mathcal{INC}(\phi, \chi) \to 1$ then $\mathcal{AMB}(\phi, \chi) \to 1$*
- *if $\mathcal{AMB}(\phi, \chi) \to 0$ then $\mathcal{INC}(\phi, \chi) \to 0$*

This proposition asserts that a highly indecisive formula is highly ambiguous, whereas a formula with low ambiguity is highly decisive.

Definition 7. *Let (ϕ, χ) be a formula in a subjective belief base B where $\chi = (b_\chi, d_\chi, u_\chi)$. The degree of dogmatism of (ϕ, χ), denoted $\mathcal{DOG}(\phi, \chi)$, is defined as:*

$$\mathcal{DOG}(\phi, \chi) = 1 - \frac{\mathcal{AMB}(\phi, \chi) + \mathcal{INC}(\phi, \chi)}{2}. \tag{6}$$

Example 8. *Let $\chi_1 = (1, 0, 0), \chi_2 = (0, 1, 0), \chi_3 = (0, 0, 1)$, and $\chi_4 = (0.5, 0.5, 0)$ be subjective opinions. The dogmatism of a formula ϕ annotated with each of these opinions is $\mathcal{DOG}(\phi, \chi_1) = 1$, $\mathcal{DOG}(\phi, \chi_2) = 1$, $\mathcal{DOG}(\phi, \chi_3) = 0.5$, $\mathcal{DOG}(\phi, \chi_4) = 0.5$.*

The most desirable state for any formula in a subjective belief base is unambiguity and decisiveness, which is achieved when all the possible belief masses are either assigned to b_χ or d_χ. A highly dogmatic formula meets these two criteria. A formula is dogmatic since it is both decisive and unambiguous, but not necessarily correct.

Definition 9. *Let B be a subjective belief base, and $\alpha \in [0, 1]$. The α-cut of B is $B_{\geq\alpha} = \{\phi \in B^* | (\phi, \chi) \in B \text{ and } \mathcal{DOG}(\phi, \chi) \geq \alpha\}$*

Two subjective belief bases are equivalent, shown as $B \equiv_s B'$, if and only if $\forall \alpha \in (0, 1]\ B_{\geq\alpha} \equiv B'_{\geq\alpha}$.

Definition 10. *[4] Let B be a subjective belief base. The degree of inconsistency of B, denoted $\mathcal{INC}(B)$, can be defined as:*

$$\mathcal{INC}(B) = max\{\alpha_i | B_{\geq\alpha_i} \text{ is inconsistent}\}. \tag{7}$$

Definition 11. *Extends [4]: Let B be a subjective belief base. A formula ϕ is said to be a consequence of B to the degree of a, denoted $B \vdash_\pi (\phi, \chi)$, where $\mathcal{DOG}(\phi, \chi) = a$, if and only if 1) $B_{\geq a}$ is consistent, 2) $B_{\geq a} \vdash \phi$ 3) $\forall a' > a, B_{\geq a'} \nvdash \phi$.*

4 Subjective Prime Implicants

Previously, weighted prime implicants have been employed to measure the quantities of conflict and agreement between two prioritized belief bases [16]. Here, this concept is further extended to support subjective belief bases.

Definition 12. *[2] A prime implicant of a belief base B is an implicant D of B such that $\forall D$ of B, $D' \not\equiv D$, and $D \not\vdash D'$.*

A prime implicant is an implicant that cannot be covered by a more general implicant. A prime implicant needs also be minimal, that is the removal of any literal from a prime implicant results in a non-implicant. For a subjective belief base such as $B = \{(\phi_1, \chi_1), ..., (\phi_n, \chi_n)\}$ where ϕ_i are clauses, a subjective implicant of B is $D = \{(\psi_1, \chi_1), ..., (\psi_n, \chi_n)\}$, such that ψ_i are literals and $D \vdash_\pi B$. Let D and D' be two subjective implicants of B, then D is believed to be subsumed by D' iff $D \neq D'$, $D'^* \subseteq D^*$ and $\forall(\psi_i, \chi_i) \in D, \exists(\psi_j, \chi_j) \in D'$ such that $\mathcal{DOG}(\psi_j, \chi_j) \leq \mathcal{DOG}(\psi_i, \chi_i)$. We can now define subjective prime implicants based on the concept of weighted prime implicants [16].

Definition 13. *Let $B = \{(\phi_1, \chi_1), ..., (\phi_n, \chi_n)\}$ be a subjective belief base, where ϕ_i are clauses. A subjective prime implicant (SPI) of B is D such that:*

- *D is a subjective implicant of B*
- *$\nexists D'$ of B such that D is subsumed by D'*

This definition shows that an SPI must not be subsumed by any other subjective implicant and should therefore be the least specific subset.

Proposition 14. *Let B be a subjective belief base. If $D_1, ..., D_n$ are all the SPIs of B, then the following must always hold:*

$$B \vdash_\pi (\phi, \chi) \text{ iff } D_i \vdash_\pi (\phi, \chi), \forall D_i \text{ of } B. \tag{8}$$

Proposition 14 is important, because it shows that the subjective prime implicants of a subjective belief base can be employed to compile the belief base itself. Each of the acquired SPIs of a given subjective belief base is normalized i.e. all literals in the form $(\neg\phi, \chi)$ is converted to (ϕ, χ') such that the subjective opinion is complemented i.e. $\chi' = (d_\chi, b_\chi, u_\chi)$. A formula $(\neg\phi, (1, 0, 0))$ is converted to $(\phi, (0, 1, 0))$. We assume that all SPIs are normalized in such a way.

Example 15. *Let $\chi_1 = (0.5, 0.3, 0.2), \chi_2 = (1, 0, 0), \chi_3 = (0.4, 0, 0.6), \chi_4 = (0.8, 0.2, 0), \chi_5 = (0, 1, 0), \chi_6 = (0, 0.4, 0.6)$ be six subjective opinions and $B_1 = \{(\phi, \chi_1), (\psi \vee \neg\mu, \chi_2)\}, B_2 = \{(\neg\phi \vee \mu, \chi_3), (\psi, \chi_4)\}$ be two subjective belief bases. The SPIs for B_1 are $C_1 = \{(\phi, \chi_1), (\psi, \chi_2)\}$ and $C_2 = \{(\phi, \chi_1), (\neg\mu, \chi_2)\}$, and the SPIs for B_2 are $D_1 = \{(\neg\phi, \chi_3), (\psi, \chi_4)\}$ and $D_2 = \{(\mu, \chi_4), (\psi, \chi_2)\}$. The normalized form of the SPIs would be $\hat{C}_1 = \{(\phi, \chi_1), (\psi, \chi_2)\}$, $\hat{C}_2 = \{(\phi, \chi_1), (\mu, \chi_5)\}$, $\hat{D}_1 = \{(\phi, \chi_6), (\psi, \chi_4)\}$ and $\hat{D}_2 = \{(\mu, \chi_4), (\psi, \chi_2)\}$.*

It is now possible to extend the given essential properties through the concept of SPIs.

Definition 16. *Let B be a subjective belief base, Suppose C is a SPI of B, then informational value of C, denoted $INFO(C)$, is defined as:*

$$INFO(C) = \sum_{(\phi,\chi)\in C} \mathcal{DOG}(\phi,\chi). \tag{9}$$

The informational value of B is defined based on the informational value of its SPIs (denoted $\mathcal{C}$):

$$\mathcal{INFO}(B) = max\{\mathcal{INFO}(C)|C \in \mathcal{C}\}. \tag{10}$$

In cases where the formulae in the belief base are completely dogmatic, the informational value of a belief base is equivalent to the size of its largest subjective prime implicant.

Definition 17. *Let B be a subjective belief base, and C and D be two of its subjective prime implicants. The quantity of conflict between these two subjective prime implicants is defined as:*

$$\mathcal{CON}(C,D) = \sum_{(\phi,\chi)\in C,(\phi,\chi')\in D} b_\chi d_{\chi'} + b_{\chi'} d_\chi. \tag{11}$$

Suppose $\mathcal{C}$ is the set of all subjective prime implicants of a subjective belief base B. The internal quantity of conflict of B, denoted $\mathcal{ICON}(B)$, is defined as:

$$\mathcal{ICON}(B) = min\{\mathcal{CON}(C,D)|C,D \in \mathcal{C}\}. \tag{12}$$

Definitions 16 and 17 allow us to define the incoherence degree of a belief base. The degree of incoherence depicts the extent of inconsistency available within the information conveyed by a single subjective belief base.

Definition 18. *Let B be a subjective belief base. The degree of incoherence of B, denoted $\mathcal{INCH}(B)$, is defined as:*

$$\mathcal{INCH}(B) = \frac{\mathcal{ICON}(B)}{\mathcal{ICON}(B) + \mathcal{INFO}(B)}. \tag{13}$$

The degree of incoherence allows us to draw an interesting conclusion about the potential value of a belief base. We can infer that a subjective belief base B is *informative* if $\mathcal{INCH}(B) < 0.5$, i.e. $\mathcal{ICON}(B) < \mathcal{INFO}(B)$. This conclusion can be reached since a subjective belief base with a degree of incoherence of higher than 0.5 is most likely to contain more conflicting/discrepant formula as opposed to consistent information.

5 Contingent Properties

The contingent properties of a subjective belief base are defined based on the context of the belief profile where a given subjective belief base is positioned; therefore, belief bases in the same belief profile have mutual effects on each others' contingent properties. In this section, we introduce several contingent properties of a subjective belief base.

Definition 19. *Let B_1 and B_2 be two subjective belief bases in a belief profile $\mathcal{SP}$. Suppose $\mathcal{C}$ and $\mathcal{D}$ are the sets of SPIs of B_1 and B_2, respectively. The quantity of conflict between the two belief bases B_1 and B_2, denoted $\mathcal{Q}_{\mathcal{CON}}(B_1, B_2)$, is defined as:*

$$\mathcal{Q}_{\mathcal{CON}}(B_1, B_2) = min\{\mathcal{CON}(C, D) | C \in \mathcal{C}, D \in \mathcal{D}\}. \tag{14}$$

Proposition 20. *Let $Dalal(w_i, w_j)$ be the Hamming distance between two models w_i and w_j of a classical formula, and $Dalal(B_1, B_1)$ be the distance between two classical belief bases defined as the minimum distance between all two models of B_1 and B_2, then*

$$\mathcal{Q}_{\mathcal{CON}}(B_1, B_2) = Dalal(B_1, B_1). \tag{15}$$

This proposition shows that the quantity of conflict between two subjective belief bases is a generalization of Dalal's distance measure for classical belief bases.

Definition 21. *Let B_1 and B_2 be two subjective belief bases, Suppose C and D are two SPIs of B_1 and B_2, respectively, then the quantity of strong agreement between C and D, denoted $Q_{\mathcal{SA}}(C, D)$, is defined as:*

$$Q_{\mathcal{SA}}(C, D) = \sum_{(\phi,\chi)\in C,(\phi,\chi')\in D} b_\chi b_{\chi'} + d_\chi d_{\chi'}. \tag{16}$$

Further suppose $\mathcal{C}$ and $\mathcal{D}$ are the set of SPIs of B_1 and B_2, respectively, then the quantity of strong agreement between B_1 and B_2 is defined as:

$$\mathcal{Q}_{\mathcal{SA}}(B_1, B_2) = max\{Q_{\mathcal{SA}}(C, D) | C \in \mathcal{C}, D \in \mathcal{D}\}. \tag{17}$$

Strong agreement focuses on the obvious similarities between the two belief bases i.e. only those formula that have been explicitly presented in both belief bases are considered while the quantity of strong agreement is measured.

Definition 22. *Let B_1 and B_2 be two subjective belief bases, suppose C and D are two SPIs of B_1 and B_2, respectively then the quantity of weak agreement between C and D, denoted $Q_{\mathcal{WA}}(C, D)$, is defined as:*

$$Q_{\mathcal{WA}}(C, D) = \sum_{\phi\in C^*\cup D^*,\phi\notin C^*\cap D^*} \mathcal{DOG}(\phi, \chi). \tag{18}$$

Now suppose $\mathcal{C}$ and $\mathcal{D}$ are the set of SPIs of B_1 and B_2, respectively, the quantity of weak agreement between B_1 and B_2 is defined as:

$$\mathcal{Q}_{\mathcal{WA}}(B_1, B_2) = max\{Q_{\mathcal{WA}}(C, D) | C \in \mathcal{C}, D \in \mathcal{D}\}. \tag{19}$$

Quantity of weak agreement measures the information that has been asserted by one belief base but has not been refuted by the other.

Proposition 23. *Let B be a subjective belief base, $\mathcal{Q}_{\mathcal{WA}}$ and $\mathcal{INFO}$ be the quantity of weak agreement and informational value measures, respectively, then:*

$$\mathcal{INFO}(B) = \mathcal{Q}_{\mathcal{WA}}(B, \varnothing). \tag{20}$$

Proposition 23 shows that the informational value of a subjective belief base is a special case of the weak agreement property. The interpretation of this is that the quality of the information in each individual belief base cannot be determined prior to its comparison with the the content of the other belief bases; therefore, no conclusions about it can be directly made. This is equivalent to a weak agreement with the empty set where the empty set represents the ignorance towards the other belief bases' contents.

Definition 24. *Let B_1 and B_2 be two subjective belief bases, and $\mathcal{Q}_{\mathcal{SA}}$, $\mathcal{Q}_{\mathcal{WA}}$, and $\mathcal{Q}_{\mathcal{CON}}$ be the quantities of strong agreement, weak agreement and conflict, respectively. The degree of conflict between two belief bases, denoted $\mathcal{D}_{\mathcal{CON}}$, is defined as follows:*

$$\mathcal{D}_{\mathcal{CON}}(B_1, B_2) = \frac{\mathcal{Q}_{\mathcal{CON}}(B_1, B_2)}{\mathcal{Q}_{\mathcal{CON}}(B_1, B_2) + \mathcal{Q}_{\mathcal{SA}}(B_1, B_2) + \lambda \mathcal{Q}_{\mathcal{WA}}(B_1, B_2)}. \tag{21}$$

where $\lambda \in [0, 1]$ is employed to weaken the effect of the quantity of weak agreement.

Definition 25. *Let B_1 be a subjective belief base, and $SP = \{B_1, ..., B_n\}$ be a belief profile. Suppose $\wp(B)$ is the number of belief bases in conflict with B, then the scope of conflict for B, denoted $\mathcal{SC}_{\mathcal{CON}}(B)$, is defined as:*

$$\mathcal{SC}_{\mathcal{CON}}(B) = \frac{\wp(B)}{n}. \tag{22}$$

It is apparent from the definition of the *scope of conflict* that if $\mathcal{SC}_{\mathcal{CON}}(B) > 0$ then $\mathcal{D}_{\mathcal{CON}} > 0$.

Example 26. *(Continues Example 15) The quantity of conflict, weak agreement, and strong agreement between B_1 and B_2 are $\mathcal{Q}_{\mathcal{CON}} = 0.2$, $\mathcal{Q}_{\mathcal{SA}} = 0.75$ and $\mathcal{Q}_{\mathcal{WA}} = 1.49$. Further, if we allow $\lambda = 0.5$, $\mathcal{D}_{\mathcal{CON}} = 0.11$.*

Proposition 27. *Let B be a subjective belief base, and $\mathcal{D}_{\mathcal{CON}}$ and $\mathcal{INCH}$ be the degree of conflict and degree of incoherence, respectively. $\mathcal{INCH}$ is a special case of $\mathcal{D}_{\mathcal{CON}}$ calculated for an individual belief base.*

The degree of conflict property reduces into the degree of incoherence when employed on a single belief base, since for an individual belief base we have: $\mathcal{Q}_{\mathcal{CON}}(B, B) = \mathcal{ICON}(B)$, $\mathcal{Q}_{\mathcal{SA}}(B) = 0$, and $\mathcal{INFO}(B)$ follows Proposition 23.

Definition 28. *Let B_1 be a subjective belief base, and $SP = \{B_1, ..., B_n\}$ be a subjective profile. Suppose η_{SP} is the number of unique formula in SP, and C is a SPI of B_1. The degree of completeness of C, denoted $COM(C)$ is defined as:*

$$COM(C) = \frac{\sum_{(\psi,\chi)\in C} \mathcal{DOG}(\psi, \chi) \times \kappa(\psi)}{\eta_{SP}}. \tag{23}$$

where $\kappa(\psi)$ is the number of clauses in SP that can be derived from ψ. Further suppose $\mathcal{C}$ is the set of all SPIs for B_1, the degree of completeness of B_1, is defined as:

$$\mathcal{COM}(B_1) = max\{COM(C) | C \in \mathcal{C}\}. \tag{24}$$

The degree of completeness shows the amount of information that a belief base can convey. If a belief base expresses all of the possible information, each with the highest possible dogmatism, it would be considered as *complete*.

Proposition 29. *Let B be a subjective belief base. Suppose B is a* ***complete*** *belief base, then B is not necessarily* ***coherent*** *i.e. $\mathcal{INCH}(B)$ does not necessarily approach 0.*

Definition 30. *(Extends Definition 28) Let B_1 be a subjective belief base, and $SP = \{B_1, ..., B_n\}$ be a subjective profile. Suppose η_{SP} is the number of unique formula in SP, and C is a SPI of B_1. The degree of comprehensiveness of C, denoted $COM_{\geq \mathcal{INC}(B_1)}(C)$ is defined as:*

$$COM_{\geq \mathcal{INC}(B_1)}(C) = \frac{\sum_{(\psi,\chi)\in C, \mathcal{DOG}(\psi,\chi)\geq \mathcal{INC}(B_1)} \mathcal{DOG}(\psi,\chi) \times \kappa(\psi)}{\eta_{SP}}. \tag{25}$$

where $\kappa(\psi)$ is the number of clauses in SP that can be derived from ψ. Further suppose $\mathcal{C}$ is the set of all SPIs for B_1, The degree of comprehensiveness of B_1, is defined as:

$$\mathcal{COM}_{\geq \mathcal{INC}(B_1)}(B_1) = max\{COM_{\geq \mathcal{INC}(B_1)}(C) | C \in \mathcal{C}\}. \tag{26}$$

We can now claim that a highly comprehensive belief base is both highly coherent and complete.

6 Ordering Belief Bases

Several researchers have approached belief merging viewing it as a multi-stage game [1,10]. The game is a competition between the information sources where the weakest of the information sources is considered as the loser of the game and should make appropriate concessions. The basic idea of the merging game is to form some sort of coalition between like-minded information sources, and penalize the information sources that are furthest away from the coalition. The rationality for such an approach under the reliability assumption for the information sources is based on Condorcet's Jury theorem [13]. The theorem argues that in most cases listening to the majority of the members of the jury is the most rational decision.

The stages of the belief merge game are known as rounds of negotiation or competition. Konieczny's belief game model [10] and Booth's belief negotiation model [1] are among the most popular frameworks for belief merging, which are defined for a pure propositional logic setting. In these models, in each round of negotiation, some of the information sources are selected by a negotiation (choice) function. The selected sources should then weaken their beliefs by making concessions using a weakening (contraction) function. In most cases the negotiation function ranks the given belief bases based on a fitness measure and selects the least significant belief base as the loser of the game that needs to make the concessions.

Definition 31. *Let B be a subjective belief base, $\mathcal{SP}$ be a belief profile of the set of all other belief base, and $\mathcal{K}$ be the set of all formula in B. The fitness measure of a subjective belief base B is a two-place function $fm : \mathcal{K} \times \mathcal{SE} \rightarrow [0, \infty)$. Given two subjective belief bases B and $B' \in \mathcal{SP}$, B is considered more competent than B' if and only if $fm_{\mathcal{SP}}(B) < fm_{\mathcal{SP}}(B')$.*

Definition 32. *Let $\mathcal{SP} = \{B_1, ..., B_n\}$ be a subjective belief profile. A fitness measure-based ordering, denoted $\prec_{fm}$, of two subjective belief bases B_i, and $B_j \in \mathcal{SP}$ is defined as:*

$$B_i \prec_{fm} B_j \quad \begin{array}{l} \textit{iff } (B_i^* \equiv \top \textit{ and } B_j^* \not\equiv \top) \\ \textit{or } (B_i^* \not\equiv \top \textit{ and } B_j^* \not\equiv \top \textit{ but } fm_{\mathcal{SP}}(B_i) < fm_{\mathcal{SP}}(B_j)). \end{array}$$

Now, the essential and contingent properties of belief bases can be employed to develop the required fitness measure in the belief base ordering process. We visit some of these properties and analyze their effect on the fitness measure's behavior.

($\mathcal{INCH}^{-1}$) [1] A fitness measure-based ordering based on the degree of incoherence of a subjective belief base would place those belief bases with the highest degree of incoherence in the lowest positions, while giving more weight to coherent belief bases.

($D^{-1}_{\mathcal{CON}}$) Belief bases can be ordered by allowing those belief bases that have the least degree of conflict with the other belief bases to possess a higher degree of importance, and those with more conflict to have lower priority and a higher chance to be chosen for making concessions.

($D^{-1}_{\mathcal{CON}}, \mathcal{COM}$) A composite measure based on $D^{-1}_{\mathcal{CON}}$ and $\mathcal{COM}$ would make sure that those belief bases that have the lowest degree of conflict with the other belief bases and at the same time are the most complete, to have the highest importance and have the lowest priority to be selected for making concessions. This measure is more complete than a single ($D^{-1}_{\mathcal{CON}}$), since in ($D^{-1}_{\mathcal{CON}}$), those belief bases that are only revealing a small amount of information and therefore have a lower chance of causing inconsistencies may receive higher degree of importance than those who provide a wealth of information but have some conflicting formula.

($D^{-1}_{\mathcal{CON}}, \mathcal{COM}, \mathcal{INCH}^{-1}$) This composite measure would assign the priority of a belief base from three perspectives: the degree of its conflict with other belief base, its completeness, and its degree of internal incoherence. This combination seems to be quite inclusive, since it analyzes all the belief bases, both individually and in the context their belief profile.

Example 33. *(Continues Example 15) Let $B_3 = \{(\phi, \chi_2)\}$ be a third subjective belief base. The degree of conflict between the belief bases is $\mathcal{D}_{\mathcal{CON}}(B_1, B_2) = 0.11$, $\mathcal{D}_{\mathcal{CON}}(B_1, B_3) = 0.16$ and $\mathcal{D}_{\mathcal{CON}}(B_2, B_3) = 0.22$. Further, the degree of completeness of the belief bases are $\mathcal{COM}(B_1) = 0.745$, $\mathcal{COM}(B_2) = 0.84$ and $\mathcal{COM}(B_3) = 0.5$. All three belief bases are internally consistent; therefore, they*

[1] The $^{-1}$ superscript shows that the inverse of the property is employed in the prioritization process.

are all completely coherent. If we assume that the composite measures are developed through averaging the constituting elements, we have the following order relationships, for $(\mathcal{INCH}^{-1})$*:* $B_1 = B_2 = B_3$*,* $(D^{-1}_{\mathcal{CON}})$*:* $B_3 < B_2 < B_1$ *and* $(D^{-1}_{\mathcal{CON}}, \mathcal{COM}) = (D^{-1}_{\mathcal{CON}}, \mathcal{COM}, \mathcal{INCH}^{-1})$*:* $B_3 < B_1 < B_2$*.*

The other properties introduced in this paper such as the scope of conflict and the degree of comprehensiveness can also be employed to develop other belief base ordering schemes.

7 Concluding Remarks

In this paper, we have explored the properties of subjective belief bases, and defined various essential and contingent properties for these belief bases. Essential properties reflect the attributes of a belief base from an individualistic viewpoint, whereas contingent properties of a belief base reveal its characteristics with regards to the rest of the peer belief bases. It has been shown that subjective belief bases are a generalization of prioritized belief bases and that such belief bases can be developed from an existing prioritized belief base. The employment of Subjective logic for annotating belief bases is useful in that it extends the Dempster-Shafer functions by explicitly defining disbelief and uncertainty, which provides more reasoning power. The properties proposed in this paper for subjective belief bases possess interesting behaviors whose individual or composite traits are shown to be useful in developing priority and ordering relationships between various belief bases, which can be helpful in the merge, revision, and management of belief bases.

References

1. Booth, R.: Social contraction and belief negotiation. Information Fusion 7(1), 19–34 (2006)
2. Cadoli, M., Donini, F.M.: A survey on knowledge compilation. AI Commun. 10(3-4), 137–150 (1997)
3. Dalal, M.: Investigations into a theory of knowledge base revision: Preliminary report. In: AAAI 1988, pp. 475–479 (1988)
4. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: Handbook of logic in AI and log. prog., pp. 439–513 (1994)
5. Higashi, M., Klir, G.J.: Measures of uncertainty and information based on possibility distributions. International Journal of General Systems 9(1), 43–58 (1982)
6. Higashi, M., Klir, G.J.: On the notion of distance representing information closeness: Possibility and probability distributions. International Journal of General Systems 9(2), 103–115 (1983)
7. Hunter, A.: Making argumentation more believable. In: AAAI 2004, pp. 269–274 (2004)
8. Jenhani, I., Ben Elouedi, Z., Benferhat, S., Mellouli, K.: Information affinity: A new similarity measure for possibilistic uncertain information. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 840–852. Springer, Heidelberg (2007)

9. Jøsang, A.: A logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(3), 212–279 (2001)
10. Konieczny, S.: Belief base merging as a game. Journal of Applied Non-Classical Logics 14(3), 275–294 (2004)
11. Konieczny, S., Lang, J., Marquis, P.: Quantifying information and contradiction in propositional logic through test actions. In: IJCAI, pp. 106–111 (2003)
12. Kroupa: Measure of divergence of possibility measures. In: 6th Workshop on Uncertainty Processing (2003)
13. List, C., Goodin, R.E.: Epistemic democracy: Generalizing the condorcet jury theorem. Journal of Political Philosophy 9(3), 277–306 (2001)
14. Liu, W.: Analyzing the degree of conflict among belief functions. Artif. Intell. 170(11), 909–924 (2006)
15. Liu, W.: Measuring conflict between possibilistic uncertain information through belief function theory. In: Almási, G.S., Caşcaval, C., Wu, P. (eds.) KSEM 2006. LNCS, vol. 4382, pp. 265–277. Springer, Heidelberg (2007)
16. Liu, W., Qi, G., Bell, D.A.: Adaptive merging of prioritized knowledge bases. Fundam. Inform. 73(3), 389–407 (2006)
17. Qi, G., Liu, W., Bell, D.: Combining multiple prioritized knowledge bases by negotiation. Fuzzy Sets and Systems 158(23), 2535–2551 (2007)
18. Qi, G., Liu, W., Bell, D.A.: Measuring conflict and agreement between two prioritized belief bases. In: IJCAI, pp. 552–557 (2005)
19. Rogova, G.L., Nimier, V.: Reliability in information fusion: Literature survey. In: International conference on Information Fusion, pp. 1158–1165 (2004)
20. Yager, R.R.: On the dempster-shafer framework and new combination rules. Inf. Sci. 41(2), 93–137 (1987)

Some Properties of Aggregated Distributions over Expected Values

David Sundgren[1], Love Ekenberg[2], and Mats Danielson[2]

[1] Dept. of Mathematics, Natural and Computer Sciences, University of Gävle, SE-801 76 Gävle, Sweden
[2] Dept. of Computer and Systems Sciences, Stockholm University and KTH, Forum 100, SE-164 40 Kista, Sweden

Abstract. Software agents and humans alike face severe difficulties in making decisions in uncertain contexts. One approach is to formalise the decision situation by means of decision theory, i.e. probabilities and utilities leading to the principle of maximising the expected utility. Expected utility is here considered as a stochastic variable; under the assumption that all utility values are equally likely, and that each vector of probability values is equally likely, the probability distribution of expected utility is calculated for two, three, and four possible outcomes. The effect of these probability distributions concentrating around the middle value is explored and its significance for making decisions.

1 Introduction

Software agents exploring new contexts continually encounter situations not explicitly addressed at compile-time. In order to act rationally, agents need to be able to reason and act according to the results of reasoning by means of some decision making procedure. The decision contexts are often characterised by uncertainties and lack of information. Likewise, human decision-makers most often face incomplete decision situations. It is in general impossible for a human decision-maker to assert precise probabilities and utilities to the consequences of decision alternatives. Further, it is questionable whether people that do not have a great deal of experience in thinking about probabilities are able to distinguish probabilities that are quite far apart [10]. And even if a human decision-maker is able to discriminate between different probabilities, very often complete, adequate and precise information is missing also in this case. Thus, for the purpose of this paper, software agents as well as humans are referred to as decision-makers.

It has since long been recognised that decision theory needs to accommodate imprecise probabilities, see e.g. [3,9,1,5,6] and also imprecise utilities [8]. But it is far from straightforward to devise an efficient decision rule that reflects the decision-maker's imprecise knowledge. If, though, the expected utilities of the decision alternatives have distributions that are concentrated enough around some central value, henceforth referred to as a centroid, it does in fact matter little how the imprecise probabilities and utilities are represented, as long as the corresponding second-order distributions have the same centroids.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 699–709, 2008.

That is, the decision-maker needs only to estimate, for each decision alternative, what the centroids of second-order distributions of probabilities and utilities would be. In other words, the decision-maker does not need the exact values, nor the entire distributions, but a reasonable estimate of the likeliest values. A form of expected utility can then be computed as the vector product $\mathbf{p} \cdot \mathbf{u}$, where $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ are vectors containing the centroids. The centroids can be given directly or be computed from interval estimates or from second-order distributions. See [4]. Also, even in situations when the generalized expected utility described above would not apply, highly concentrated distributions of expected utility would in many cases minimize the problem of overlapping intervals of expected utility.

Intuitively, the desired effect of belief concentration of expected utility would be expected to be stronger with increased width and height of the decision tree. If this is indeed the case, paradoxically, the more complicated a decision situation is, the more justification there would be to use an extremely easily implementable and efficient decision rule. Here we support this intuition by producing the probability density functions for expected utility in decision trees with depth one and width two, three and four. Results of this kind have not been published before. Although this concentration effect is to be expected when considering the Law of Large Numbers, we believe that it is illuminating to consider the actual probability density functions in order to establish in which situations the use of centroids is justified. Further, the distributions of expected utility are quite hard to calculate and time consuming to simulate, so pre-computed pdf:s of expected utility such as those presented in this paper would be useful.

Consider a decision situation where we have a decision alternative that can lead to n different possible outcomes E_i. Event E_i occurs with probability p_i and has utility value u_i. In this paper, we consider the problem: what is the probability that the expected utility $z = \sum_{i=1}^{n} p_i u_i$ is within a given interval? We find that when the number of outcomes, n, grows, the interval within which a given fraction of the belief mass, say 75%, is found becomes more narrow.

We assume for simplicity that the decision-maker has no additional information on either probabilities or utilities other than that the probabilities are indeed probabilities, i.e. non-negative and have to sum to one, and that the utilities are normalised to the interval $[0, 1]$. In other words, the decision-maker has no reason to believe that any one of the n possible outcomes is either more probable or more desirable than any other, and there are no limitations on the probabilities other than that probabilities of mutually exclusive events have to sum to one. Corresponding results for more than four possible outcomes and with sharper underlying distributions for probabilities and utilities are a matter for future research.

Although this assumption may seem unrealistic, the conclusions that can be drawn from this simplified situation can be expected to hold to an even higher degree when probabilities and utilities are more sharply specified. Please note that we are concerned only with situations that may be modelled by a decision

tree of depth one; the n possible outcomes do not in turn lead to other outcomes with their own probabilities and utilities.

2 The Probability Density Function $h_n(z)$

Under the assumptions made above, we study the expected utility z,

$$\sum_{i=1}^{n} p_i u_i = z$$

where p_i is the probability and u_i the utility of outcome E_i. The set of outcomes E_i is a partition of the sample space, so $\sum_{i=1}^{n} p_i = 1$.

The problem is, what is the probability that the expected utility z lies within a given interval? Let us denote the probability density function of the expected utility z by $h_n(z)$. As for the probabilities, we have the *uniform belief* assumption; every point in the $n-1$-dimensional polytope formed by $p_i \geq 0$ and $\sum_{i=1}^{n-1} p_i \leq 1$ is considered to be equally likely. By [11], we have that the probability density function of probability p_i is $f(p_i) = (n-1)(1-p_i)^{n-2}$. Further, the utility u_i has a probability density function $g(u_i) = 1, 0 \leq u_i \leq 1$.

2.1 Proof by Geometry

Let us first consider $h_2(z)$, $p_1u_1 + (1-p_1)u_2 = z$. Geometrically, $h_2(z)$ is the area of the surface $p_1u_1 + (1-p_1)u_2 = z$, cut by the box $(u_1, u_2, p_1) \in [0,1]^3$. Likewise, the cumulative distribution

$$H_2(z) = \Pr(p_1u_1 + (1-p_1)u_2 \leq z)$$

(not to be confused with the harmonic number H_n) may be interpreted geometrically as the volume enclosed by the surface $p_1u_1 + (1-p_1)u_2 = z$ and the coordinate planes.

We proceed by calculating $H_2(z)$ as a volume. With p_1 fixed, we have an area in $[0,1]^2$ where $p_1u_1 + (1-p_1)u_2 \leq z$, see Figure 1, where we discern two different cases. Without loss of generalisation we assume that $0 \leq z \leq 1/2$.

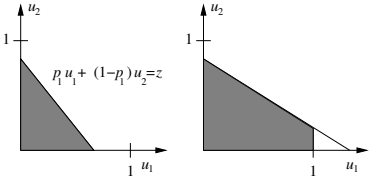


Fig. 1. The area enclosed by $p_1u_1 + (1-p_1)u_2 = z$ and $u_1, u_2 \in [0,1]$, where p_1 is fixed. To the left where $p_1 \geq z$ and to the right where $p_1 < z$.

This area is a trivial application of a *box spline*, see [2]; when $p_1 \geq z$, we simply have a triangle with area $\frac{z^2}{2p_1(1-p_1)}$ and when $p_1 < z$ the surface is a polytope with area $\frac{z^2}{2p_1(1-p_1)} - \frac{(z-p_1)^2}{2p_1(1-p_1)}$. Thus

$$H_2(z) = \int_z^{1-z} \frac{z^2}{2p_1(1-p_1)}\,dp_1 + 2\int_0^z \frac{z^2-(z-p_1)^2}{2p_1(1-p_1)}\,dp_1 = (1-z)^2\ln(1-z) - z^2\ln(z) + z\,.$$

Differentiating $H_2(z)$ with respect to z gives

$$h_2(z) = -2(1-z)\ln(1-z) - 2z\ln(z)\,.$$

In Figure 2 we see an example of h_2 as area.

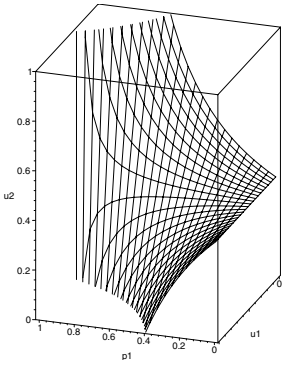


Fig. 2. Surface $p_1u_1 + (1-p_1)u_2 = 0.4$ with area $h_2(0.4) = -1.2\ln(0.6) - 0.8\ln(0.4)$

Similarly, we use three-dimensional box splines to calculate the cumulative distribution $H_3(z)$;

$$H_3(z) = \int_z^{1-2z}\int_z^{1-z-p_1} \frac{z^3}{6p_1p_2(1-p_1-p_2)}\,dp_2\,dp_1 + \int_0^z\int_z^{1-z-p_1} \frac{z^3-(z-p_1)^3}{2p_1p_2(1-p_1-p_2)}\,dp_2\,dp_1 + \int_0^z\int_0^{z-p_1} \frac{z^3-(z-p_1)^3-(z-p_2)^3+(z-p_1-p_2)^3}{2p_1p_2(1-p_1-p_2)}\,dp_2\,dp_1 =$$
$$-2(1-z)^3\,\mathrm{Li}_2(1-z) + 2z^3\,\mathrm{Li}_2(z) + z^3\ln^2(z) - (1-z)^3\ln^2(1-z) + 3(1-z)^2\ln(1-z) - 3z^2\ln(z) + 2z(1-z)(\ln(z)-\ln(1-z)) + 2((1-z)^3 - z^3)\,\mathrm{Li}_2(1) + z\,,$$

where $\mathrm{Li}_n(z)$ is the *polylogarithm of order* n, a generalisation of the natural logarithm. $\mathrm{Li}_n(z)$ defined as the series $\sum_{k=1}^{\infty} \frac{z^k}{k^n}$ for complex numbers n and z where $|z| < 1$. For z with $|z| \geq 1$, $\mathrm{Li}_n(z)$ may be defined recursively as $\int_0^z \frac{\mathrm{Li}_{n-1}(z)}{t}\, dt$, where $\mathrm{Li}_1(z) = -\ln(1-z)$. See e.g. [7].

Again, taking the derivative with respect to z produces the probability density function;

$$h_3(z) = \frac{d}{dz} H_3(z) =$$
$$6(1-z)^2 \mathrm{Li}_2(1-z) + 6z^2 \mathrm{Li}_2(z) - 6(1-z)\ln(1-z) - 6z\ln(z) +$$
$$3(1-z)^2 \ln^2(1-z) + 3z^2 \ln^2(z) - 6(z^2 + (1-z)^2)\,\mathrm{Li}_2(1)\,.$$

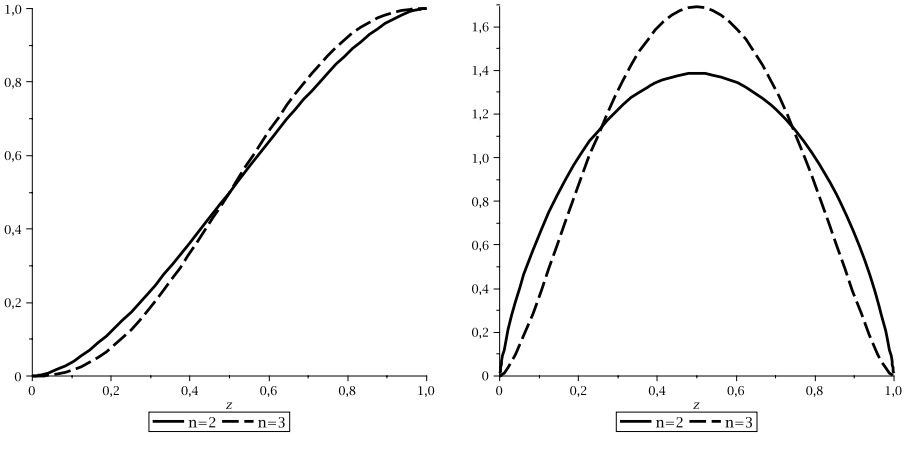
Geometrically speaking, $h_3(z)$ is the area of the five-dimensional hyper surface

$$p_1 u_1 + p_2 u_2 + (1 - p_1 - p_2) u_3 = z\,,$$

$(u_1, u_2, u_3) \in [0,1]^3, p_1, p_2 \geq 0, p_1 + p_2 \leq 1$.

In Figure 3, we see the graphs of the cumulative distribution functions $H_2(z)$ and $H_3(z)$ and the probability density functions $h_2(z), h_3(z)$, respectively.

The box spline method appears not to be practical for calculating $H_4(z)$, instead we proceed by computing $h_4(z)$ with probability theoretical reasoning.



(a) Cumulative distributions $H_2(z)$ (solid line) and $H_3(z)$ (dashed)

(b) Probability density functions $h_2(z)$ (solid line) and $h_3(z)$ (dashed)

Fig. 3. Distribution of expected utility with two or three possible outcomes

2.2 Proof by Probability

Next, we examine $h_n(z)$ from a probability-theoretic and recursive perspective. Consider $H_{n+1}(z)$ in relation to $H_n(z)$;

$$H_{n+1}(z) = \Pr\left(\sum_{i=1}^{n+1} p_i u_i \leq z\right) = \Pr\left(\sum_{i=1}^{n} p_i u_i \leq z - p_{n+1}u_{n+1}\right).$$

Replacing z with $z - p_{n+1}u_{n+1}$ in $H_n(z)$ and letting $\sum_{i=1}^{n} p_i = 1 - p_{n+1}$ instead of $\sum_{i=1}^{n} p_i = 1$ lets us through a linear transformation describe the probability that $\sum_{i=1}^{n} p_i u_i \leq z - p_{n+1}u_{n+1}$ given p_{n+1} and u_{n+1} as

$$H_n\left(\frac{z - p_{n+1}u_{n+1}}{1 - p_{n+1}}\right),$$

for $p_{n+1}u_{n+1} \leq z \leq 1 - p_{n+1}(1 - u_{n+1})$. Observe that $p_i, i = 1, \ldots, n$ are no longer probabilities, but non-negative numbers summing to $1 - p_{n+1}$.

The density functions for u_{n+1} and p_{n+1}, respectively are again $g(u_{n+1}) = 1$ and

$$f(p_{n+1}) = n(1 - p_{n+1})^{n-1}.$$

This means, with p_{n+1} abbreviated as p and u_{n+1} as u, that

$$H_{n+1}(z) = \int_z^1 \int_0^{z/u} n(1-p)^{n-1} H_n\left(\frac{z - pu}{1-p}\right) dp\, du + \int_0^z \int_0^{\frac{1-z}{1-u}} n(1-p)^{n-1} H_n\left(\frac{z - pu}{1-p}\right) dp\, du.$$

With the variable change $x = \frac{z-pu}{1-p}$ we get

$$H_{n+1}(z) = \int_z^1 nH_n(x) \int_0^z \frac{(z-u)^n}{(x-u)^{n+1}}\, du\, dx - \int_0^z nH_n(x) \int_z^1 \frac{(u-z)^n}{(u-x)^{n+1}}\, du\, dx.$$

Taking the derivative with respect to z, and using symmetry, and that $H_n(1-x) = 1 - H_n(x)$, H_n being a cumulative distribution gives, with $H_1(x) = x$ (one possible, hence certain, outcome), $h_2(z) = f_2(z) + f_2(1-z)$, where

$$f_2(z) = z\ln(1-z) - \lim_{\epsilon \to 0}\left(\int_0^{z-\epsilon} \frac{x(1-z)}{(z-x)(1-x)}\, dx + z\ln(\epsilon)\right),$$

and $h_3(z) = f_3(z) + f_3(1-z) - 2$, where

$$f_3(z) = 2H_2(z)\ln(1-z) - \lim_{\epsilon \to 0}\left(\int_0^{z-\epsilon} \frac{2H_2(x)(1-z)^2}{(z-x)(1-x)^2}\, dx + 2H_2(z)\ln(\epsilon)\right).$$

$H_2(z), h_2(z)$ and $h_3(z)$ are, of course, the same functions as above.

And, finally, we are ready for $h_4(z)$;

$$h_4(z) = f_4(z) + f_4(1-z) - \frac{9}{2}\,.$$

where

$$f_4(z) = 3H_3(z)\ln(1-z) - \lim_{\epsilon\to 0}\left(\int_0^{z-\epsilon} \frac{3H_3(x)(1-z)^3}{(z-x)(1-x)^3}\,dx + 3H_3(z)\ln(\epsilon)\right).$$

$$\begin{aligned} h_4(z) = {} & 24(z^3+(1-z)^3)\,\mathrm{Li}_2(z)\,(\ln(1-z)-\ln(z)) - \\ & 24(z^3+(1-z)^3)\,\mathrm{Li}_3\left(\frac{z-1}{z}\right) + 4(1-z)^3\left(\ln^3(z)-\ln^3(1-z)\right) + \\ & 6(5z^2-8z+3)\ln(1-z)\,(\ln(1-z)-2\ln(z)) - \\ & 36(1-2z)\,\mathrm{Li}_2(z) + 12(z^3+2(1-z)^3)\ln(z)\ln(1-z)^2 - \\ & 12(z^3+2(1-z)^3)\ln(z)^2\ln(1-z) - 12(1-z)\ln(1-z) - 12z\ln(z) - \\ & 6z(2-5z)\ln(z)^2 + 4\pi^2(2z^3+(1-z)^3)\ln(z) - 4\pi^2 z^3\ln(1-z) - 2\pi^2 z(5z-2)\,. \end{aligned}$$

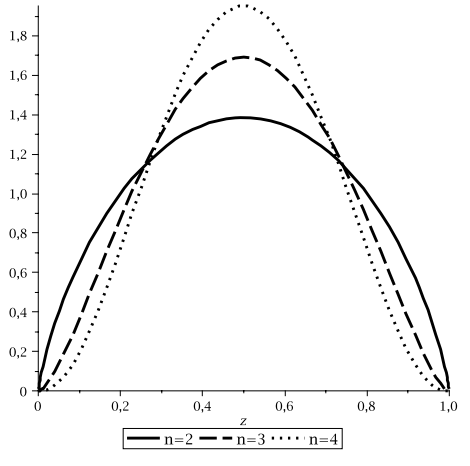
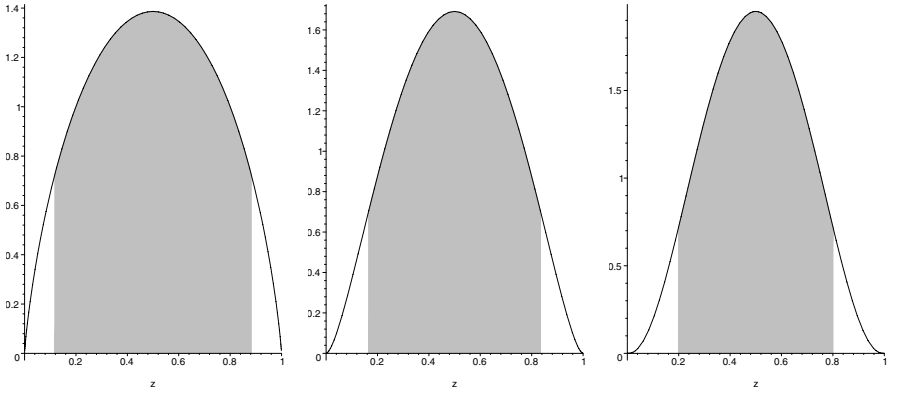


Fig. 4. $h_2(z)$, $h_3(z)$ and $h_4(z)$

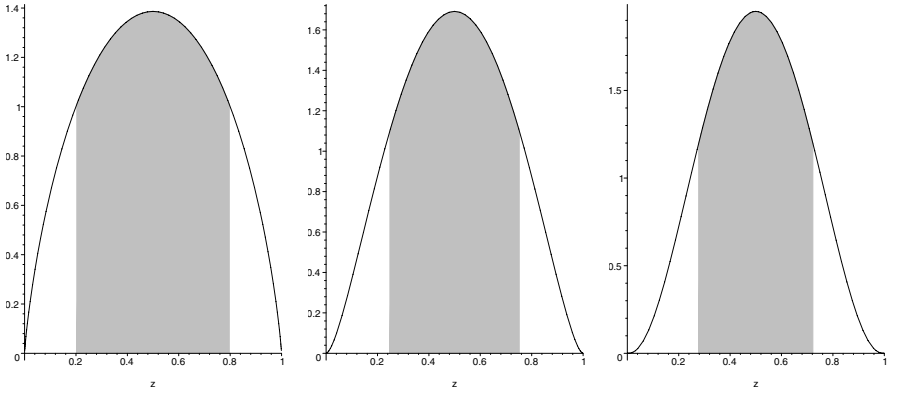
3 Concentration of Belief

One of the most important aims of this paper is to show evidence of increasing concentration around the centroid with growing n. In Figure 4 we see graphs



(a) $h_2(z), 0.118 \leq z \leq 0.882$ (b) $h_3(z), 0.166 \leq z \leq 0.834$ (c) $h_4(z), 0.198 \leq z \leq 0.802$

Fig. 5. 90 % of the belief mass marked in grey



(a) $h_2(z), 0.203 \leq z \leq 0.797$ (b) $h_3(z), 0.248 \leq z \leq 0.752$ (c) $h_4(z), 0.278 \leq z \leq 0.722$

Fig. 6. 75 % of the belief mass marked in grey

of the functions h_2, h_3 and h_4 computed above. We show in Figures 5 and 6 in which intervals 90% and 75%, respectively, of the belief mass is located for $n = 2, 3$ and 4.

We note a moderate but distinct concentration of belief already when $n \leq 4$ and probabilities and utilities being unrestricted.

As of yet, it is an open question whether a higher degree of concentration occurs when probabilities and utilities are under constraints by e.g. intervals or second-order distributions.

4 Comparison with Beta Distributions

The shapes of h_2, h_3 and h_4 suggests a similarity to beta distributions. Here we will explore how close these probability density functions are to those of beta distributions with corresponding mean and variance. The mean of $h_n(z)$ is obviously $1/2$; the variance of $h_2(z)$ is

$$\int_0^1 (z-1/2)^2 h_2(z)\,dz = \frac{1}{18}\,,$$

the variance of $h_3(z)$ is

$$\int_0^1 (z-1/2)^2 h_3(z)\,dz = \frac{1}{24}\,,$$

and the variance of $h_4(z)$ is

$$\int_0^1 (z-1/2)^2 h_4(z)\,dz = \frac{1}{30}\,.$$

That is, the variance is $\frac{1}{6(n+1)}$ at least for $n = 2, 3$ and 4.

This implies that the beta distributions that have the same mean and variance as h_2, h_3 and h_4, respectively, are the symmetric beta distributions with parameters $\left(\frac{3n+1}{4}, \frac{3n+1}{4}\right)$, for $n = 2, 3$ and 4. In Figure 7 we see a graph of the difference

$$h_n(z) - \frac{(z(1-z))^{\frac{3(n-1)}{4}}}{\mathrm{B}\left(\frac{3n+1}{4}, \frac{3n+1}{4}\right)}$$

for $n = 2, 3$ and 4.

At least for $n \leq 4$, the numerical difference is quite small. With the exception of $n = 2$ near $z = 0$ and $z = 1$, the absolute value of the difference is less than 0.01. However, for $z = 1/2$, the pdf of the beta distribution is larger than h_n, and the difference grows with n. Whether the difference converges or not is a topic for further research.

Assuming that the difference for, say, $n \leq 7$ is small enough to be of little practical importance, we can expect approximately the same numerical values for h_5, h_6 and h_7 and the pdf:s of Beta distributions with parameters $\left(\frac{3n+1}{4}, \frac{3n+1}{4}\right)$ for $n = 5, 6$ and 7, respectively. If this indeed is the case, the graphs in Figure 8 of the pdf:s of these Beta distributions with 90 % of the area under the curves marked in grey also show the concentration of belief for the corresponding h_n:s.

Whether the variance of the probability density function h_n is $\frac{1}{6(n+1)}$ for n larger than four is also a question for further research.

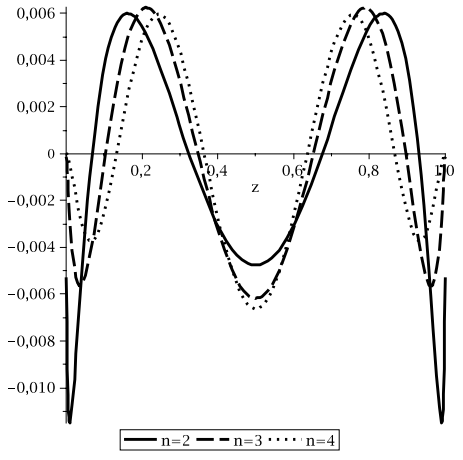
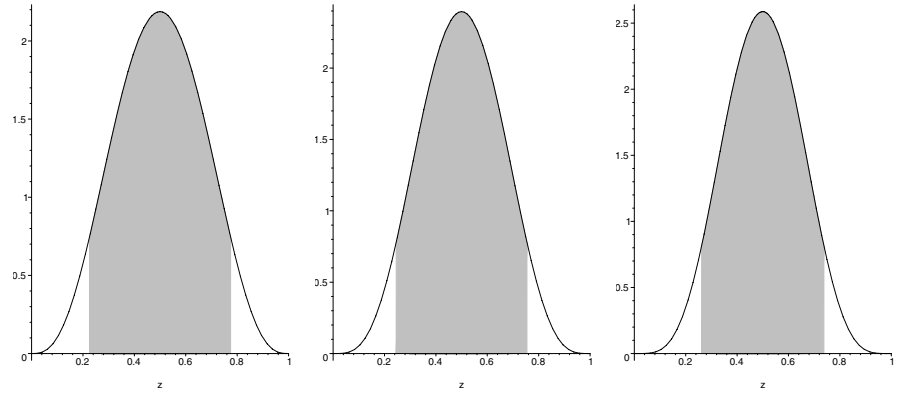


Fig. 7. Difference between $h_n(z)$ and the pdf:s of the corresponding beta distributions



(a) $n = 5$,
$0.225 \leq z \leq 0.775$

(b) $n = 6$,
$0.246 \leq z \leq 0.754$

(c) $n = 7$,
$0.262 \leq z \leq 0.738$

Fig. 8. 90 % of the belief mass marked in grey

5 Conclusions

We have gained some insights into the distribution of expected utility; at least for small values of n ($n \leq 4$), we have established the probability density function of expected utility in the case of n possible outcomes when no restrictions are laid upon either probabilities or utilities (uniform belief assumption). Through comparison with Beta distributions, we have an approximation of the behaviour of distribution of expected utility for somewhat larger values of n.

There is some concentration of belief under the simplifying assumptions and no reason to believe that this effect would be any lesser when probabilities or utilities are more sharply specified through e.g. intervals or second order distributions. Exactly what happens with the distribution of expected utility under such circumstances, and even under the uniform belief assumption with an arbitrary number of possible outcomes n, will be topics for further research.

These results are of both theoretical and practical interest. Theoretical since distributions of expected utility have hitherto not been known, and practical since knowing the actual distribution will aid decision-makers in choosing a decision rule. Further, distributions such as those computed here may be used in maximizing the expected utility.

References

1. Choquet, G.: Theory of capacities. Ann. Inst. Fourier 5, 131–295 (1954)
2. DeConcini, C., Procesi, C.: The algebra of the box spline (2006), http://www.citebase.org/abstract?id=oai:arXiv.org:math/0602019
3. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. Annals of Mathematical Statistics xxxviii, 325–399 (1967)
4. Ekenberg, L., Thorbiörnson, J.: Second-order decision analysis. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(1), 13–38 (2001)
5. Huber, P.J.: The case of choquet capacities in statistics. Bulletin of the International Statistical Institute 45, 181–188 (1973)
6. Huber, P.J., Strassen, V.: Minimax tests and the neyman-pearsons lemma for capacities. Annals of Statistics 1, 251–263 (1973)
7. Lewin, L.: Polylogarithms and Associated Functions. North-Holland, Amsterdam (1981)
8. Nau, R.F.: Uncertainty aversion with second-order utilities and probabilities. Management Science 52(1), 136–145 (2006)
9. Shafer, G.: A Mathematical theory of evidence. Princeton University Press, Princeton (1976)
10. Shapira, Z.: Risk Taking: A Managerial Perspective. Russel Sage Foundation, Thousand Oaks (1995)
11. Sundgren, D., Danielson, M., Ekenberg, L.: Some second order effects on interval based probabilities. In: Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, pp. 848–853. AAAI Press, Menlo Park (2006)

Generators of Fuzzy Operations for Hardware Implementation of Fuzzy Systems

Imre J. Rudas[1], Ildar Z. Batyrshin[2], Antonio Hernández Zavala[3], Oscar Camacho Nieto[3], László Horváth[1], and Luis Villa Vargas[3]

[1] Budapest Tech
rudas@bmf.hu, horvath.laszlo@nik.bmf.hu
[2] Mexican Petroleum Institute
batyr1@gmail.com
[3] Mexican Polytechnic Institute – Computer Research Centre
antonioh@hotmail.com, {oscarc,lvilla}@cic.ipn.mx

Abstract. The problem of effective hardware implementation of parametric operations of fuzzy systems is studied in this paper. The methods of generation of parametric classes of fuzzy conjunctions and disjunctions by means of introduced generators and basic operations are considered. Several types of generators of parametric fuzzy operations simple for hardware implementation are proposed. Examples of hardware implementation of proposed parametric operations are presented.

Keywords: Fuzzy logic, fuzzy processor, fuzzy system, conjunction, generator.

1 Introduction

Hardware implementation of fuzzy systems plays an important role in many industrial applications of fuzzy logic based intelligent systems in control systems, signal processing, pattern recognition, expert systems, decision making etc [1-3]. To provide a flexibility of fuzzy systems in modeling real processes it is important to implement families of parametric fuzzy logic operations that can be tuned to obtain better simulation results of fuzzy system [4-6]. On-board and real-time applications of fuzzy systems require faster processing speed of fuzzy hardware. To achieve these goals it is desirable to propose fuzzy parametric operations that can be effectively implemented in hardware. In [7], a new method of generation of parametric family of fuzzy conjunctions that have simple hardware implementation was proposed and FPGA digital implementation of new parametric conjunctions using Xilinx tools was considered.

In this paper we propose a new approach to generation of parametric families of fuzzy conjunctions simple for hardware implementation. This approach is based on the use of simple parametric generators together with *min-max*, Lukasiewicz and drastic *t*-norms and *t*-conorms [8-9] that have effective digital implementation. Several simple parametric classes of generators are proposed and examples of parametric conjunctions generated by means of these generators and basic *t*-norms and *t*-norms are considered. The proposed approach is based on the results of papers [4-6] where

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 710–719, 2008.

the methods of generation of parametric classes of fuzzy operations simple for tuning in fuzzy systems were introduced. But most of parametric operations proposed in [4-6] use product operations or exponential functions that do not admit efficient hardware implementation. In this paper we introduce new parametric families of fuzzy operations that have efficient digital hardware implementation.

The paper has the following structure. Section 2 gives the basic definitions of fuzzy conjunction and disjunction operations and considers the methods of generation of these operations. In Section 3 a digital representation of fuzzy operations is considered and several simple parametric families of generators of fuzzy conjunctions are proposed. The methods of a hardware implementation of proposed parametric operations are considered in Section 4. In Conclusions we discuss obtained results and future directions of research.

2 Basic Definitions

In binary logic, the set of true values contains only two elements $L = \{0,1\}$, the negation $\neg$, conjunction $\wedge$ and disjunction $\vee$ operations are defined as follows:

$$\neg 0 = 1, \qquad \neg 1 = 0, \tag{1}$$

$$0 \wedge 0 = 0,\ 1 \wedge 0 = 0,\ 0 \wedge 1 = 0, \qquad 1 \wedge 1 = 1, \tag{2}$$

$$0 \vee 0 = 0, \qquad 1 \vee 0 = 1,\ 0 \vee 1 = 1,\ 1 \vee 1 = 1. \tag{3}$$

In fuzzy logic, the set of true values (called membership values) usually contains continuum number of elements $L = [0,1]$ and negation, conjunction and disjunction operations are defined as functions $N: L \to L$, $T: L \times L \to L$ and $S: L \times L \to L$. Generalization of properties of this operations (1)-(3) on the set $L = [0,1]$ can be done by different ways [10-12, 6, 5, 8,13].

Fuzzy involutive negation is defined by the following axioms:

$$N(0) = 1, \qquad N(1) = 0, \tag{4}$$

$$N(x) \geq N(y), \qquad \text{if } x \leq y, \qquad \textit{(antimonotonicity)} \tag{5}$$

$$N(N(x)) = x. \tag{6}$$

Generally fuzzy negation is defined by (4)-(5). Most popular fuzzy conjunction and disjunction functions are called t-norms and t-conorms [8] and defined by commutativity, associativity and (7)-(9) axioms:

$$T(x,1) = x, \qquad T(1,y) = y, \qquad \textit{(boundary conditions)} \tag{7}$$

$$S(x,0) = x, \qquad S(0,y) = y, \qquad \textit{(boundary conditions)} \tag{8}$$

$$T(x,y) \leq T(u,v), \quad S(x,y) \leq S(u,v), \quad \textit{if } x \leq u,\ y \leq v. \quad \textit{(monotonicity)} \tag{9}$$

As it was pointed out in [4-6] associativity and commutativity are very useful properties of these operations in mathematical models of fuzzy logic but in industrial applications of fuzzy systems these properties do not play important role. For this reason fuzzy conjunction and disjunction can be defined only by axioms (7)-(9). Another reason for such definition of these operations is that due to associativity parametric t-norms and t-conorms usually have complicated forms for tuning in fuzzy systems and for hardware implementation.

From (7)-(9) it follows:

$$T(x,0) = 0,\ T(0,y) = 0, \tag{10}$$

$$S(x,1) = 1,\ S(1,y) = 1, \tag{11}$$

and, hence, the properties (2)-(3) are fulfilled in points $\{0,1\}$ for fuzzy conjunctions and disjunctions defined by (7)-(9).

Consider the following three pairs of simplest t-norms and t-conorms:

$$T_M(x,y) = min\{x,y\},\ \ (minimum), \qquad S_M(x,y) = max\{x,y\},\ \ (maximum), \tag{12}$$

$$T_L(x,y) = max\{x+y-1,\ 0\},\quad S_L(x,y) = min\{x+y,\ 1\},\qquad (Lukasiewicz), \tag{13}$$

$$T_D(x,y) = \begin{cases} 0, & if\ \ (x,y) \in [0,1)\times[0,1) \\ \min(x,y), & otherwise \end{cases},\qquad (drastic\ product), \tag{14}$$

$$S_D(x,y) = \begin{cases} 1, & if\ \ (x,y) \in (0,1]\times(0,1] \\ \max(x,y), & otherwise \end{cases} \qquad (drastic\ sum). \tag{15}$$

These three pairs of conjunction and disjunction operations will be considered as basic conjunction and disjunction operations. These operations have efficient hardware implementation [7] and they will be used further for generation of new parametric classes of conjunction and disjunction operations. Note that the product *t*-norm $T_P(x,y) = x \cdot y$, and the probabilistic sum $S_P(x,y) = x + y - x \cdot y$ have less effective digital hardware implementation than basic operations (12)-(15) due to presence of product operation in their definition [7].

Pairs of *t*-norms and *t*-conorms considered above can be obtained one from another by means of negation $N(x) = 1 - x$ as follows:

$$S(x,y) = N(T(N(x),N(y))), \qquad T(x,y) = N(S(N(x),N(y))). \tag{16}$$

If N is an involution then these relations also can be used for obtaining from a given fuzzy conjunction T (or disjunction S) corresponding disjunction S (conjunction T). In this case the obtained pair of fuzzy conjunction and disjunction operations together with involution N will constitute a De Morgan triple (N,T,S). Further we will consider the methods of generation of fuzzy conjunctions. Fuzzy disjunctions can be obtained from them by means of (16).

It can be shown [5] that fuzzy conjunctions and disjunctions satisfy the following inequalities:

$$T_D(x,y) \le T(x,y) \le T_M(x,y) \le S_M(x,y) \le S(x,y) \le S_D(x,y), \tag{17}$$

so, min, max and drastic operations define the borders for fuzzy conjunctions and disjunctions.

Several methods of generation of simple parametric fuzzy operations suitable for tuning in fuzzy systems have been proposed in [4-6]. These methods are based on the formula:

$$T(x,y)= T_2(T_1(x,y),S(x,y)), \tag{18}$$

where T_2 and T_1 are conjunctions and S is a pseudo-disjunction. Recall that *pseudo-conjunctions* T and *pseudo-disjunctions* S are defined in [5] as functions satisfying (9)-(11), respectively. Suppose $h,g_1,g_2:L\to L$ are non-decreasing functions called generators. Pseudo-disjunction S can de generated as follows:

$$S(x,y)= S_1(g_1(x),g_2(y)), \tag{19}$$

$$S(x,y)= g_1(S_1(x,y)), \tag{20}$$

$$S(x,y)= S_2(S_1(x,y),h(y)), \tag{21}$$

where g_1 and g_2 satisfy conditions: $g_1(1)= g_2(1)= 1$. As pseudo-disjunctions S in (18) one can use basic disjunctions or pseudo-disjunctions obtained from basic disjunctions by recursive application of (18)-(21). Several classes of parametric conjunctions simple for tuning in fuzzy systems were proposed in [4-6] but most of them have not efficient hardware implementation because they are based on product operation or exponential function.

In the present paper we study the problem of construction of wide class of fuzzy parametric conjunction and disjunction operations with effective hardware implementation. We propose several classes of generators simple for hardware implementation. These generators together with basic conjunctions and disjunctions can be used to generate parametric classes of fuzzy conjunctions with effective hardware implementation.

3 Digital Generators of Fuzzy Operations

Suppose it is used m bits in digital representation of membership values. Then different membership values can be represented by 2^m numbers from the set $L_D= \{0,1,2,\ldots, 2^m-1\}$. Denote the maximal membership value $I = 2^m-1$. This value will represent the full membership corresponding to the value 1 in a traditional set of membership values $L= [0,1]$. All definitions and properties of fuzzy operations (1)-(21) from the previous section can be transformed into the digital case by replacing the set of membership values $L= [0,1]$ by $L_D= \{0,1,2,\ldots, 2^m-1\}$ and maximal membership value 1 by I. Also in definition of drastic product and sum in (14), (15) we need to replace intervals [0,1) and (0,1] by the sets $\{0,1,2, \ldots ,2^m-2\}$ and $\{1,2,\ldots, 2^m-1\}$, respectively. For graphical representation of digital generators and fuzzy operations we will use below $m= 4$ bits, with the following set of digital membership values $L_D=\{0,1,2,\ldots,14,15\}$ and maximal membership value I= 15.

We need to use in (19)-(21) generators that have effective hardware implementation. For this reason we propose the simplest generators depending on one parameter p. We suppose that parameter p can change from 0 till I. Suppose 0.5(-) and 0.5(+) denote digital representations of the membership value 0.5. They are defined as follows: $0.5(-) = 2^{m-1}-1$, $0.5(+) = 2^{m-1}$. For example, for m= 4 bits we have $I = 2^m-1 = 15$, 0.5(-) = 7 and 0.5(+) = 8. We propose the following simplest generators that have efficient hardware implementation:

$$g(x) = \begin{cases} 0, & if \quad x < p \\ I, & otherwise \end{cases}, \qquad (0I\_step) \tag{22}$$

$$g(x) = \begin{cases} 0, & if \quad x \le p \\ x, & otherwise \end{cases} \qquad (0\text{-}diag) \tag{23}$$

$$g(x) = \begin{cases} x, & if \quad x \le p \\ I, & otherwise \end{cases} \qquad (I\text{-}diag) \tag{24}$$

$$g(x) = \begin{cases} 0, & if \quad x \le p \\ x, & if \quad p < x < I - p \\ I, & otherwise \end{cases} \qquad (0I\text{-}diag) \tag{25}$$

$$g(x) = \begin{cases} x, & if \quad x \le p \\ p, & if \quad p < x \le 0.5(-) \\ I - p, & if \quad 0.5(+) \le x < I - p \\ I, & otherwise \end{cases} \qquad (p\text{-}diag) \tag{26}$$

$$g(x) = \begin{cases} \max(x - p, 0), & if \quad x \le 0.5(-) \\ \min(x + p, I), & otherwise \end{cases} \qquad (bound\text{-}dif\text{-}sum) \tag{27}$$

Fig.1. shows these generators in digital representation with 4 bits for parameter value p = 3 corresponding to value 0.2 in interval [0,1] of true values.

Fig.2 contains an example of parametric fuzzy conjunction obtained by (18) and (19) by means of basic fuzzy conjunctions and disjunction operations: T_1= minimum, T_2= Lukasiewicz conjunction, S_1= maximum, and with generators: g_1= p-diag, g_2= 0I-step. The value of the parameters p in generator p-diag is equal to 0.2 in [0,1] scale of true values. For digital representation with 4 bits and I = 15 it corresponds to value p= 3. Parameter p in generator g_2= 0I-step equals to 0.7 in [0,1] scale or 11 in 4 bits representation.

Fig. 3 contains an example of parametric fuzzy conjunction obtained by (18) and (20) by means of generator 0I-diag and basic fuzzy conjunctions and disjunction

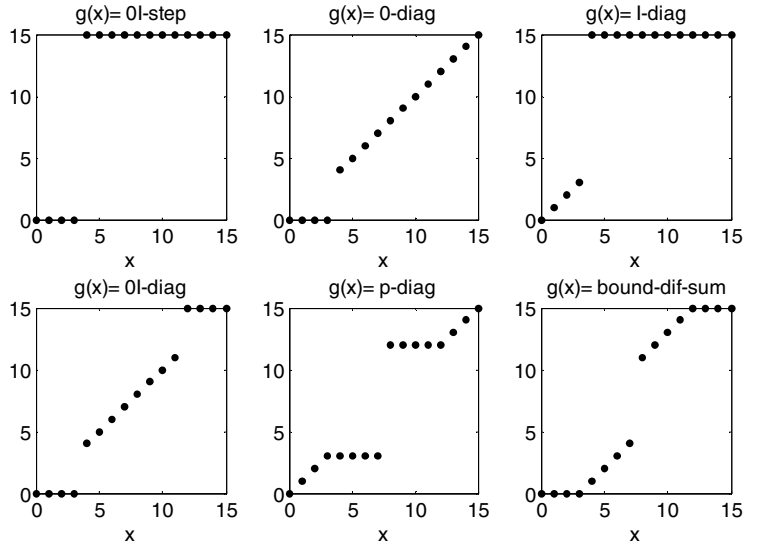


Fig. 1. Generators with parameter p= 3 in digital representation with 4 bits

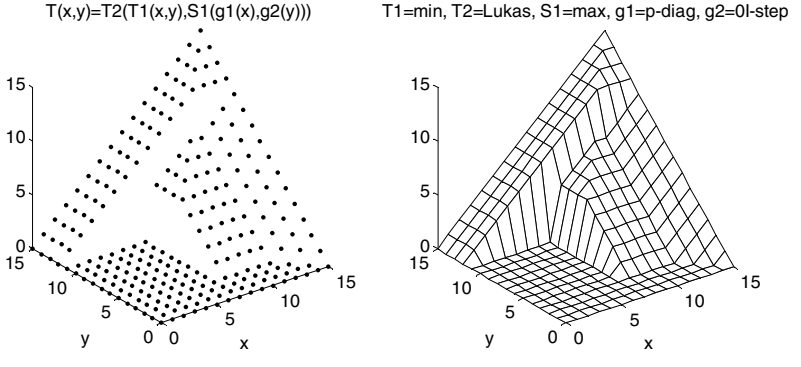


Fig. 2. Example of parametric fuzzy conjunction obtained by (18) and (19). On the left it is shown a shape of conjunction function with digital representation with 4 bits and on the right it is presented the same shape as a mesh figure.

operations: T_1= Lukasiewicz conjunction, T_2= drastic conjunction, S= maximum. The value of the parameter p in generator 0I-diag is equal to 0.2 in [0,1] scale of true values or 3 in 4 bits representation.

Fig. 4 contains an example of parametric fuzzy conjunction obtained by (18) and (21) by means of generator h= bound-dif-sum and basic fuzzy conjunctions and disjunction operations: T_1= minimum, T_2= Lukasiewicz conjunction, S_1= maximum, S_2= drastic disjunction. The shape of obtained fuzzy conjunction corresponds to the value of the parameter p in generator bound-dif-sum equal to 0.4 in [0,1] scale of true values and 6 in digital representation with 4 bits.

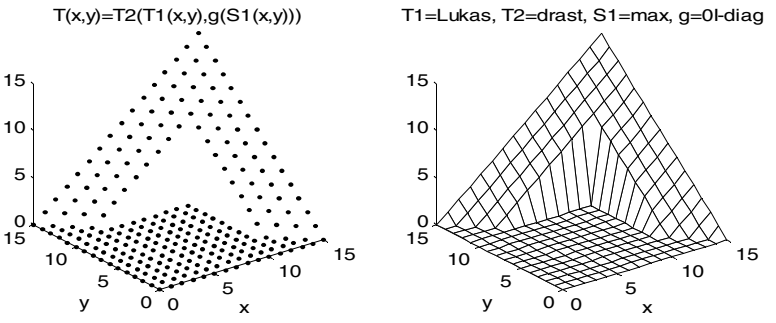


Fig. 3. Example of parametric fuzzy conjunction obtained by (18) and (20)

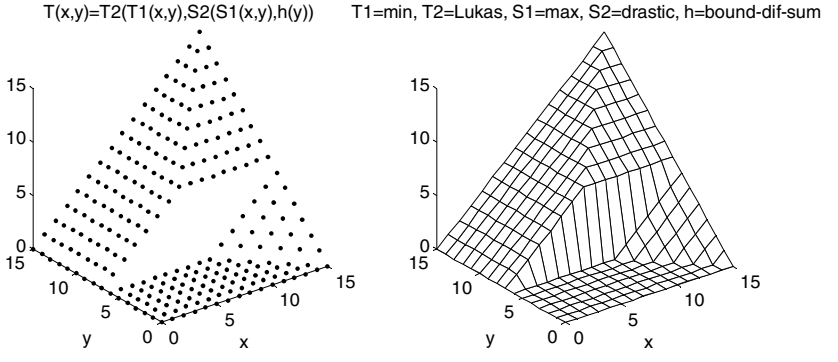


Fig. 4. Example of parametric fuzzy conjunction obtained by (18) and (21)

4 Digital Hardware Implementation of Parametric Fuzzy Operations

In this section we introduce basic functions that can be used for generation and hardware implementation of generators and basic fuzzy conjunctions and disjunctions considered in previous sections. To apply (18)-(21) we need to use functions that have

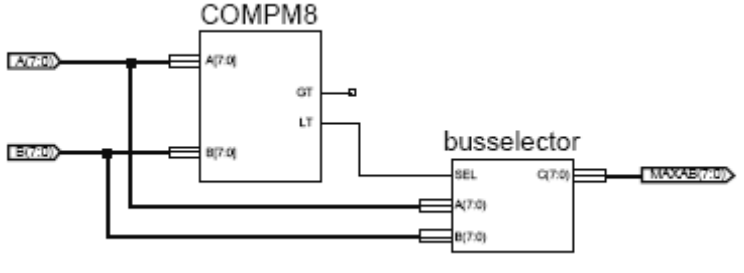


Fig. 5. Hardware implementation of maximum operation

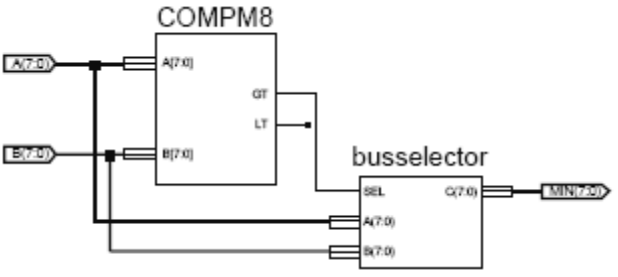


Fig. 6. Hardware implementation of minimum operation

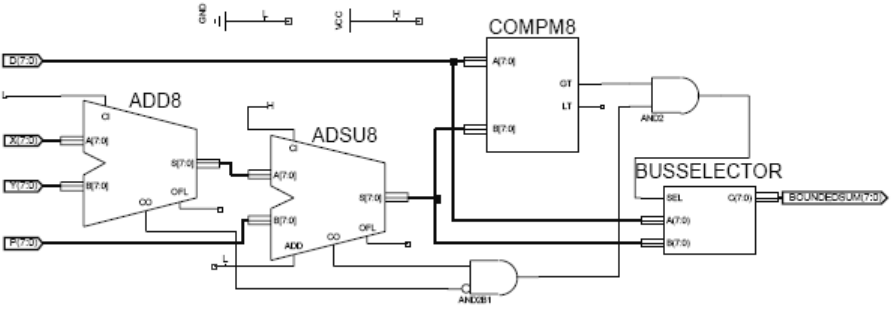


Fig. 7. Hardware implementation of bounded sum operation

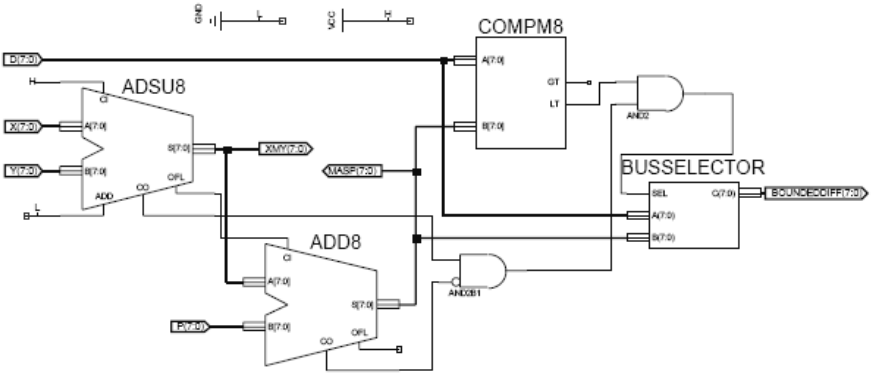


Fig. 8. Hardware implementation of bounded difference operation

effective digital hardware implementation. These functions we can obtain as a superposition of the following *basic functions* which have effective digital hardware implementation: constant, identity, minimum, maximum, bounded sum: $F(x,y)= min(1,x+y)$, and bounded difference: $F(x,y)= max(0,x-y)$. Also comparison operations can be used as basic functions in superposition of considered above functions. It is clear that the basic conjunctions and disjunctions (12)-(15) and introduced generators (22)-(27) can be considered as compositions of these basic functions. The circuits corresponding to basic functions are realized using Xilinx tools for FPGA design.

Basic logical gates, comparator, adder and subtractor [14-16] are used to construct the circuits. The circuit of 8-bits digital hardware implementation of the main basic functions are shown in Fig. 5-8. These circuits can be used as bricks in construction of generators and fuzzy conjunction and disjunction operations considered in this paper.

5 Conclusions

The main contribution of the paper is the following. The problem of effective digital hardware implementation of fuzzy systems with parametric conjunction and disjunction operations is formulated and its solution is outlined. Most of known parametric fuzzy conjunction and disjunction operations have not effective hardware implementation because they use product, division, exponentiation or other operations that have not effective hardware implementation. For example, minimum and maximum are simple comparisons between two values, which on digital circuitry is easy to implement, but a product is much more complex because of the following reasons [15,16]:

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.

A proposed new approach is based on the use of simple generators that can be composed with simple fuzzy conjunction and disjunction operations. Several classes of such parametric generators are proposed and examples of parametric fuzzy conjunctions obtained by means of these generators are presented. Further, the basic functions that can be used in construction of generators and basic conjunction and disjunction functions are considered. The circuits defining digital hardware implementation of these functions are presented. These circuits can be used as bricks in hardware implementation of generators and fuzzy conjunctions and disjunctions. The obtained results can be extended in several directions. First, the proposed approach can be extended on generation of parametric t-norms and t-conorms that have efficient digital hardware implementation. 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 that can be used as components of applied intelligent systems in control, pattern recognition and decision making.

Acknowledgments. The research work was partially supported by CONACYT, NKTH, OTKA, 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. Kandel, A., Langholz, G.: Fuzzy Hardware: Architectures and Applications. Kluwer Academic Publishers, Dordrecht (1997)
4. Batyrshin, I., Kaynak, O., Rudas, I.: Generalized Conjunction and Disjunction Operations for Fuzzy Control. In: EUFIT 1998, 6th European Congress on Intelligent Techniques & Soft Computing, vol. 1, pp. 52–57. Verlag Mainz, Aachen (1998)
5. Batyrshin, I., Kaynak, O.: Parametric Classes of Generalized Conjunction and Disjunction Operations for Fuzzy Modeling. IEEE Transactions on Fuzzy Systems 7, 586–596 (1999)
6. Batyrshin, I., Kaynak, O., Rudas, I.: Fuzzy Modeling Based on Generalized Conjunction Operations. IEEE Transactions on Fuzzy Systems 10, 678–683 (2002)
7. Batyrshin, I., Hernández Zavala, A., Camacho Nieto, O., Villa Vargas, L.: Generalized Fuzzy Operations for Digital Hardware Implementation. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 9–18. Springer, Heidelberg (2007)
8. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer, Dordrecht (2000)
9. Jang, J.-S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence. Prentice-Hall International, Englewood Cliffs (1997)
10. Fodor, J.: A New Look at Fuzzy Connectives. Fuzzy Sets Syst. 57, 141–148 (1993)
11. Batyrshin, I.: On the Structure of Involutive, Contracting and Expanding Negations. Fuzzy Sets and Syst. 139, 661–672 (2003)
12. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainty, and Information. Prentice-Hall, NJ (1988)
13. De Cooman, G., Kerre, E.E.: Order Norms on Bounded Partially Ordered Sets. J. Fuzzy Math. 2, 281–310 (1994)
14. Tocci, R.J., Widmer, N.S., Moss, G.L.: Digital Systems: Principles and Applications, 9th edn. Prentice Hall, Englewood Cliffs (2003)
15. Patterson, D.A., Hennessy, J.L.: Computer Organization and Design: The Hardware / Software Interface, 2nd edn. Morgan Kaufmann, San Francisco (1998)
16. Zargham, M.R.: Computer Architecture: Single and Parallel Systems. Prentice-Hall, Englewood Cliffs (1995)

Quantitative Possibilistic Networks: Handling Interventions and Ascribing Causality

Salem Benferhat[1] and Salma Smaoui[2]

[1] CRIL, Université d'Artois, France
[2] LARODEC, ISG de Tunis, Tunisia

Abstract. Causality notion in the possibilistic framework has not been widely studied, despite its importance in context of poor or incomplete information. In this paper, we first propose an approach for handling interventions in quantitative possibilistic networks. The main advantage of this approach is its ability to unify treatments of both observations and interventions through the propagation process. We then propose a model based on quantitative possibilistic networks for ascribing causal relations between elements of the system by presenting some of their properties. Using such graphical structures allows to provide a more parcimonious inference process (comparing to the possibilistic model based on System P) that both accepts interventions and observations.

1 Introduction

Possibilistic networks [1,2] make reference to the possibility theory. Possibilistic networks have been applied in many domains such as fault detection. Possibilistic networks use inference mechanisms, also known as propagation process, to evaluate influences between elements of a system. Two kinds of possibilistic networks exist: qualitative and quantitative possibilistic networks directly related to the two possibilistic reasoning frameworks. In this paper, we focus on the quantitative ones. Several propagation algorithms are proposed for possibilistic networks. As long as experimentation conditions remain unchanged, applying the propagation process depend on statistic analysis.

Causal analysis is more informative in the sense that it allows not only to evaluate the possibility of an event in static conditions but also to predict those events dynamics when conditions of experimentation evolve. This ability also includes anticipation of actions (or interventions) effects and identification of causes (i.e. causality ascriptions).

Unfortunately, causality notion in the possibilistic framework remains unexplored despite its importance. In fact, a very few papers have addressed this issue [3,4].

Pearl and his colleagues [5,6] have introduced a way for handling interventions in Bayesian networks using an operator, called 'do'. This 'do' operator has been adapted in possibilistic networks framework to make the difference between spontaneous changes and modalities [4].

Intervention is a useful notion when talking about causality and especially causality ascriptions. After providing approaches to deal with interventions in possibilistic causal networks, we are interested in question like 'is the event A a cause of the event B?' In [3], authors proposed a possibilistic non-monotonic model for causality ascriptions

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 720–731, 2008.

which satisfies the requirements of System P. This approach handles a purely qualitative knowledge and does not provide any algorithm to attribute causality relationships. It also cannot distinguish between possible facts and external interventions.

The main contribution of the paper is to propose a model for ascribing causality that also handles interventions by integrating the 'do' operator in the causal process. The model is developed in quantitative possibilistic networks (which has not been considered before). We derive natural postulates for causal relations, namely causality, facilitation and justification. Moreover, the new models provide algorithms to identify the kind of relationships between elements of the system.

The rest of the paper is organized as follows: Section 2 gives a brief background on possibility theory and possibilistic networks. An approach for handling interventions in possibilistic networks is presented in Section 3 using the possibilistic counterpart of the 'do' operator. Section 4 presents a possibilistic model for ascribing causal relations using quantitative possibilistic networks and some properties of these existing relations.

2 Backgrounds on the Possibilistic Frameworks

Let $V = \{A_1, A_2, ..., A_n\}$ be a set of variables. D_{A_i} denotes the finite domain associated with the variable A_i. For the sake of simplicity, and without lost of generality, variables considered here are assumed to be binary. a_i denotes any of the two instances of A_i and $\neg a_i$ represents the other instance of A_i. $\varphi, \psi, ..$ denote propositional formulas obtained from V and logical connectors $\wedge$ (conjunction),$\vee$ (disjunction),$\neg$ (propositional negation). $\top$ and $\bot$, respectively, denote tautologies and contradictions.

$\Omega = \times_{A_i \in V} D_{A_i}$ represents the universe of discourse and ω, an element of Ω, is called an *interpretation*. It is either denoted by tuples $(a_1, ..., a_n)$ or by conjunctions $(a_1 \wedge ... \wedge a_n)$, where a_i's are respectively instance of A_i's. In the following, $\models$ denotes the propositional logic satisfaction. $\omega \models \varphi$ means that ω is a model of φ.

2.1 Possibility Theory

Issued from fuzzy sets theory, possibility theory is seen as a simple and natural model for handling uncertain data.

One of the basic elements of possibility theory is the notion of possibility distribution π which is a mapping from Ω to the interval $[0, 1]$. The degree $\pi(\omega)$ represents the compatibility of ω with available pieces of information. By convention, $\pi(\omega) = 1$ means that ω is totally possible, and $\pi(\omega) = 0$ means that ω is impossible. When $\pi(\omega) > \pi(\omega')$, ω is a preferred to ω' for being the real state of the world. A possibility distribution π is said to be normalized if there exists at least one interpretation which is fully consistent with available pieces of information. More formally, $\exists \omega \in \Omega, \ \ \pi(\omega) = 1$.

An event φ is a propositional formula obtained from V and logical connectors $\wedge$,$\vee$,$\neg$. Uncertainty on an event $\varphi \subset \Omega$ can be described by two dual measures: possibility measure Π and necessity measure N.

Considering a possibility distribution π, the possibility measure of a formula φ is as follows: $\Pi(\varphi) = \max\{\pi(\omega) : \omega \models \varphi\}$, where $\Pi(\varphi)$ represents the possibility degree that a model of φ exists in the real word. This measure evaluates the consistency level of φ with information encoded by π.

A necessity measure of a formula φ is defined as follows: $N(\varphi) = 1 - \Pi(\neg\varphi)$ which corresponds to the certainty degree associated with φ from available pieces of information encoded by π.

The main difference between quantitative and qualitative possibilistic reasoning is the used operator when merging and revising pieces of information [7]. The $\min$ operator is used for combining information in the qualitative reasoning while the quantitative one uses the $product$ operator. The qualitative possibilistic framework has been widely studied. In this paper, we focus on quantitative reasoning which has strong relationships with infinitesimal probabilities and ordinal conditional functions. Formally, given a possibility distribution π and two events ϕ and ψ, $\Pi(\phi|\psi)$ is defined in the quantitative framework by:

$$\Pi(\phi|\psi) = \frac{\Pi(\phi \wedge \psi)}{\Pi(\psi)} \quad (1)$$

if $\Pi(\psi) > 0$.

2.2 Possibilistic Networks

Possibilistic networks, denoted ΠG, can be viewed as the counterparts of probabilistic bayesian networks [8]. A standard possibilistic network consists of:

- A graphical component which is a directed acyclic graph (DAG). Nodes correspond to variables and edges represent dependence relations.
- A numerical component which quantify uncertainty by means of local conditional possibility distributions at the level of each node in the context of its parents. If A is a root then the a priori distribution must satisfy $\max(\pi(a), \pi(\neg a)) = 1$. If A has parents then conditional distributions must satisfy $\max(\pi(a|u_A), \pi(\neg a|u_A)) = 1$, for each $u_A \in D_{U_A}$, where D_{U_A} is the cartesian product of domains of variables which are parents of A.

Possibilistic networks are compact representations of possibility distributions. The joint possibility distribution is obtained using the 'chain rule' defined by:

$$\pi_{\Pi G}(a_1, ..., a_n) = \prod_{i=1..n} \pi(a_i \mid u_{A_i}), \quad (2)$$

where a_i is an instance of A_i and $u_{A_i} \subseteq \{a_1, ..., a_n\}$ is an element of the cartesian product of domains associated with variables U_{A_i} which is the parent's set of A_i.

3 Handling Interventions in Possibilistic Causal Networks

In this paper, we propose to introduce the 'do' operator for inference and causal ascriptions. Using the 'do' operator in the possibilistic framework is not surprising since this operator was firstly introduced [5] in ordinal conditional functions of Spohn [9] which is a model for uncertainty representation closely related to infinitesimal probabilities and possibility theory. A possibilistic causal network is a possibilistic network where arrows represent not only dependence relationships but also cause/effect relationships. When background knowledge is provided by an expert the possibilistic network is generally a possibilistic causal network.

In this section, we briefly discuss different approaches to represent interventions in possibilistic causal networks by proposing the possibilistic counterpart of the 'do' operator [5,6]. As it is the case in the probabilistic framework, interventions are seen as modalities on variables. A simple intervention, denoted $do(A_i = a_i)$ or $do(a_i)$, that forces the variable A_i to take the value a_i is called atomic. For the sake of simplicity, all interventions considered in the following sections are atomic. The same results can be easily extended to general interventions. Graphically, an intervention can be represented different ways in possibilistic causal networks.

The effect of an intervention $do(a_i)$ on a variables set Y is denoted $\Pi(Y|do(a_i))$. In possibilistic causal networks, an intervention on a variable A_i implies that our beliefs on the parents set U_i of A_i, here given by means of possibilities, will not be affected. This modality is interpreted by the deletion (also called mutilation) of links between U_i and A_i. The rest of the graph remains intact. The resulting graph is denoted ΠG_{mut}. In this graph, all parents in U_i, which represented the direct causes of A_i, become independent of this variable.

The effect of such intervention on the rest of the graph is given by applying conditioning after observing $A_i = a_i$ in the mutilated graph. Hence, the variable concerned by the intervention becomes independent of its parents and the event $A_i = a_i$ becomes certain ($\pi_{\Pi G_{mut}}(a_i) = 1$).

A different but equivalent approach to represent intervention in possibilistic causal networks consists of augmenting the graph by adding extra nodes. This representation is an adaptation of Pearl's approach [6] proposed in the bayesian networks framework. Hence, a parent node DO_i, is added to the variable A_i concerned by the intervention. The parents set of A_i is augmented by DO_i: $U_i' = U_i \cup DO_i$. This extra node describes the behavior od A_i and takes its value in $\{\{do_{a_i} : \forall a_i \in D_{A_i}\} \cup do_{i-noact}\}$ where $do_{i-noact}$ is the state of non-intervention and do_{a_i} describes the situation where A_i is forced to take the value a_i. The augmented graph is denoted ΠG_{aug}.

The new conditional possibility distribution at the level of A_i is given, $\forall a_i' \in D_{A_i}$, by:

$$\pi(a_i'|u_i') = \begin{cases} 1 & if\ a_i' = a_i \\ \pi(a_i'|u_i)\ if\ DO_i = do_{i-noact} \\ 0 & else \end{cases} \tag{3}$$

where u_i is the projection of u_i' on U_i.

The effect of an intervention $do(a_i)$ on the rest of the system is computed by conditioning the augmented graph ΠG_{aug} by the certain piece of information $DO_i = do_{a_i}$.

The two ways of representing intervention in possibilistic causal networks are equivalent.

Proposition 1. *Let ΠG be a possibilistic causal network. Let ΠG_{mut} be the mutilated graph resulting of the intervention $do(A_i = a_i) = do(a_i)$ on the variable A_i of the graph. ΠG_{aug} represents the augmented graph constructed from ΠG by adding the extra node DO_i to the variable A_i concerned by the intervention. $\forall \omega, \forall a_i \in D_{A_i}$,*

$$\begin{aligned} &i)\ \pi(\omega) = \pi_{\Pi G_{aug}}(\omega|DO_i = do_{i-noact}) \\ &ii)\ \pi(\omega|do(a_i)) = \pi_{\Pi G_{mut}}(\omega|a_i) = \pi_{\Pi G_{aug}}(\omega|do_{a_i}) \end{aligned} \tag{4}$$

where $\pi_{\Pi G_{mut}}$ (resp. $\pi_{\Pi G_{aug}}$) is the possibility distribution associated with ΠG_{mut} (resp. ΠG_{aug}) obtained by ΠG which is a compact representation of π.

4 Ascribing Causality in Possibilistic Causal Networks

Halpern and Pearl [10] propose a model allowing identification of 'actual causes'. The model distinguishes between *endogenous* and *exogenous* variables. Assigned values of endogenous variables are governed by structural equations, whereas exogenous variables are assumed to be known and out of control. Only endogenous variables can be causes or be caused.The apparent lack of selective power of this model may be considered a weakness, as an event is very easily designated as a cause of another.

Possibilistic non-monotonic approach proposed in [3] is an alternative to the model based on structural equations proposed by Halpern and Pearl [10]. In fact, incomplete or insufficient information may render the construction of structural equations impossible. In this case, the model based on non-monotonic consequence relations allows to answer requests by handling qualitative knowledge. The inference engine based on System P is very cautious. Assuming the property of Rational Monotony provides a more psychologically plausible representation of background knowledge and default inference. The purpose of this approach is not to provide the real cause but to predict causal relations that an agent may assign using available information. The non-monotonic consequence relations $a_i \mid\sim a_j$, expressing the fact that 'In context a_i, a_j is generally true', can be simply encoded in the possibilistic framework [11]. Namely, $a_i \mid\sim a_j$ means that $N(a_j|a_i) > 0$ which is equivalent to $\Pi(a_i \wedge a_j) > \Pi(a_i \wedge \neg a_j)$ where a_i is an instance of the variable or the event A_i. Nevertheless, this model suffers from its inability to distinguish between possible facts (i.e. observations) and external manipulations (i.e. interventions). Moreover, no algorithm is proposed in the possibilistic framework for ascribing causality relations. Lastly, this approach has been only defined in the qualitative setting.

In this paper, we propose to adapt this model for the quantitative possibilistic framework. Thanks to possibilistic causal networks, this model provides a computational tool for causal relations identification and also allows to handle interventions. Interventions are treated by introducing the 'do' operator in this model.

The following section provides causal relations ascriptions in the quantitative framework when evidence only concerns observations.

4.1 Ascribing Causal Relations in Presence of Simple Observations

Causality is a complementary notion of independence. In a purely quantitative framework, a_j is independent of a_i in the context a_k if and only if $\Pi(a_j|a_k a_i) = \Pi(a_j|a_k)$ or equivalently if and only if $N(a_j|a_k a_i) = N(a_j|a_k)$. Independence between two events can be interpreted in two different ways:

- if $N(a_j|a_i a_k) = N(a_j|a_k) = 0$ then a_i does not inform about a_j;
- if $N(a_j|a_i a_k) = N(a_j|a_k) > 0$ then a_i is absolutely independent of a_j.

If events are not independent, namely, if $N(a_j|a_k a_i) \neq N(a_j|a_k)$, then these events are called relevant [12] which means that a causal relation exists between them. Hence, a_i is relevant to a_j in the context a_k if and only if $N(a_j|a_k a_i) \neq N(a_j|a_k)$. This section considers definitions of independence, causality, facilitation, justification, confirmation and attenuation.

In the following, $N(a_j|a_i)$ **is defined as** $1 - \Pi(\neg a_j|a_i) = 1 - \frac{\Pi(\neg a_j \wedge a_i)}{\Pi(a_i)}$**.**

Definition 1. *Assume that we have a sequence of events where an event a_j is rejected and becomes accepted after observing the event a_i. We say that a_i is the cause of a_j. Formally, an event a_i **causes** another event a_j if $N(\neg a_j) > 0$ and $N(a_j|a_i) > 0$.*[1]

Example 1. *Let us suppose that the background knowledge of an agent is given by means of a possibilistic network (see figure 1) containing the following variables: A_{back} means that 'A is driving behind the driver B', ACC means that 'an accident happens', A_{att} for 'A is paying attention', A_b for 'A brakes in time', B_b means that 'the driver B brakes suddenly', WR for 'the road is wet', A_{cl} for 'A changes the lane', CL means that 'it is possible to change the lane' and A_{st} means that 'the driver A succeeds in stopping in time'. The local possibility distributions are given in tables 1, 2 and 3.*

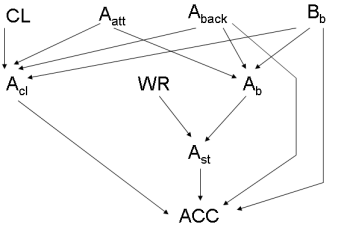


Fig. 1. The possibilistic causal network

Table 1. Local possibility distributions at the level of CL, A_{att}, A_{back}, B_b and WR

$\pi(CL)$		$\pi(A_{att})$		$\pi(A_{back})$		$\pi(B_b)$		$\pi(WR)$	
0	1	0	0.5	0	1	0	1	0	0.4
1	0.8	1	1	1	1	1	0.6	1	1

In the context $A_{back} = 1$, $ACC = 1$ is rejected: $\Pi(ACC = 1 \wedge A_{back} = 1) = 0.6 < \Pi(ACC = 0 \wedge A_{back} = 1) = 1$. If $B_b = 1$ happens, $ACC = 1$ becomes accepted: $\Pi(ACC = 1 \wedge A_{back} = 1 \wedge B_b = 1) = 1 > \Pi(ACC = 0 \wedge A_{back} = 1 \wedge B_b = 1) = 0.8$. The agent concludes that $B_b = 1$ is the cause of $ACC = 1$ in the context $A_{back} = 1$.

Definition 2. *Assume that we have a sequence of events where an event a_j is rejected and becomes ignored after observing the event a_i. We say that a_i facilitates the occurrence of a_j. Formally, an event a_i **facilitates** the occurrence of another event a_j if $N(\neg a_j) > 0$ and $N(a_j|a_i) = N(\neg a_j|a_i) = 0$*

Example 2. *Let us consider the same background knowledge given in the example 1. In the initial context, $B_b = 1$ cancels our beliefs on ACC. In fact, initially, $ACC = 1$ is rejected: $\Pi(ACC = 1) = 0.6 < \Pi(ACC = 0) = 1$. After observing the event $B_b = 1$, ACC becomes ignored: $\Pi(ACC = 1 \wedge B_b = 1) = \Pi(ACC = 0 \wedge B_b = 1) = 1$. We conclude that B_f facilitates the occurrence of $ACC = 1$.*

[1] Definitions 1–3 are exactly the same as the ones proposed in [3], however here conditioning is defined on quantitative setting. This has its importance especially for characterizing attenuation.

Table 2. Local possibility distributions at the level of A_{cl} and ACC

$\pi(A_{cl}\|CL, A_{att}, A_{back}, B_b)$		
$CL, A_{att}, A_{back}, B_b$	$A_{cl}=1$	$A_{cl}=0$
0, 0, 0, 0	0.01	1
1, 0, 0, 0	0.01	1
0, 1, 0, 0	0.01	1
1, 1, 0, 0	0.2	1
0, 0, 1, 0	0.01	1
1, 0, 1, 0	1	1
0, 1, 1, 0	0.1	1
1, 1, 1, 0	0.5	1
0, 0, 0, 1	0.01	1
1, 0, 0, 1	1	1
0, 1, 0, 1	0.01	1
1, 1, 0, 1	0.5	1
0, 0, 1, 1	0.1	1
1, 0, 1, 1	0.8	1
0, 1, 1, 1	0.2	1
1, 1, 1, 1	1	0.1

$\pi(ACC\|A_{back}, B_b, A_{cl}, A_{st})$		
$A_{back}, B_b, A_{cl}, A_{st}$	$ACC=1$	$ACC=0$
0, 0, 0, 0	0.1	1
1, 0, 0, 0	0.2	1
0, 1, 0, 0	1	1
1, 1, 0, 0	1	0.01
0, 0, 1, 0	0.01	1
1, 0, 1, 0	0.01	1
0, 1, 1, 0	0.2	1
1, 1, 1, 0	0.4	1
0, 0, 0, 1	0.1	1
1, 0, 0, 1	0.01	1
0, 1, 0, 1	0.01	1
1, 1, 0, 1	0.4	1
0, 0, 1, 1	0.01	1
1, 0, 1, 1	0.01	1
0, 1, 1, 1	0.2	1
1, 1, 1, 1	0.01	1

Table 3. Local possibility distributions at the level of A_b and A_{st}

$\pi(A_b\|A_{att}, A_{back}, B_b)$		
A_{att}, A_{back}, B_b	$A_b=1$	$A_b=0$
0, 0, 0	0.1	1
1, 0, 0	0.2	1
0, 1, 0	0.3	1
1, 1, 0	0.5	1
0, 0, 1	0.15	1
1, 0, 1	0.4	1
0, 1, 1	1	1
1, 1, 1	1	0.1

$\pi(A_{st}\|WR, A_b)$		
WR, A_b	$A_{st}=1$	$A_{st}=0$
0, 0	0.3	1
1, 0	0.1	1
0, 1	1	0.2
1, 1	0.4	1

Definition 3. *Assume that we have a sequence of events where an event a_j is ignored and becomes accepted after observing the event a_i. We say that a_i justifies the occurrence of a_j. Formally, an event a_i* ***justifies*** *the occurrence of another event a_j if $N(a_j) = N(\neg a_j) = 0$ and $N(a_j|a_i) > 0$;*

Example 3. *Let us consider the same background knowledge given in the example 1. Initially, A_{back} is ignored: $\Pi(A_{back} = 1) = \Pi(A_{back} = 0) = 1$. After observing the event $CL = 1$, the agent accepts $A_{back} = 1$: $\Pi(A_{back} = 1 \wedge CL = 1) = 1 > \Pi(A_{back} = 0 \wedge CL = 1) = 0.5$. We conclude that $CL = 1$ justifies the occurrence of A_{back}.*

In addition to different forms of influence considered above, the confirmation and attenuation fully make sense in the quantitative framework. This is not the case in the qualitative framework where attenuation can never happen.

- An event a_i **confirms** the acceptation of another event a_j if $N(a_j) > 0$ and $N(a_j|a_i) > N(a_j)$;
- An event a_i **attenuates** the acceptation of another event a_j if $N(a_j) > 0$ and $N(a_j) > N(a_j|a_i) > 0$.

Note that other possible cases may be recovered simply switching $\neg a_j$ and a_j or a_j and a_i.

Example 4. *Let us consider the possibilistic causal network given in the figure 2. The binary variable A_{att} means that the driver A is paying attention when driving and the variable ACC means that an accident happens. Local possibility distributions are given in Tab. 4. Joint possibility distribution after observing $A_{att} = 1$ are given in Tab. 5.*

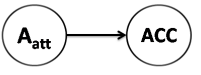


Fig. 2. Example of possibilistic causal network

Table 4. Local possibility distributions

$A_{att} = 1$	0.4
$A_{att} = 0$	1

	$A_{att} = 1$	$A_{att} = 0$
$ACC = 1$	0.6	0.2
$ACC = 0$	1	1

Table 5. Joint possibility distribution after observing $A_{att} = 1$

A_{att} ACC	$\Pi(A_{att}, ACC)$	$\Pi(A_{att}, ACC \mid A_{att} = 1)$
1 1	0.4	0.6
1 0	0.4	1
0 1	0.2	0
0 0	1	0

We note that $\Pi(ACC = 1) = 0.4$ and that $\Pi(ACC = 1|A_{att} = 1) = 0.6$. This implies that $\Pi(ACC = 1) < \Pi(ACC = 1|A_{att} = 1) < 1$ and $\Pi(ACC = 0) > \Pi(ACC = 0|A_{att} = 1) > 0$. We conclude that $A_{att} = 1$ attenuates the acceptation of $ACC = 0$.

The following propositions give some properties of existing causal relations in the quantitative possibilistic framework.

Proposition 2. *(Causality and facilitation) Let a_i, a_j and a_k be events. We have:*

1. *a_i causes a_j if and only if $\Pi(\neg a_j \wedge a_i) < \Pi(a_i) = \Pi(a_j \wedge a_i) \leq \Pi(a_j) < 1$.*
2. *a_i facilitates a_j if and only if $0 < \Pi(\neg a_j \wedge a_i) = \Pi(a_j \wedge a_i) = \Pi(a_i) \leq \Pi(\neg a_j) < \Pi(a_j) = 0$.*
3. *If a_i causes or facilitates a_j then $\Pi(a_i) < \Pi(\neg a_i)$.*

The following gives some properties regarding remaining concepts:

Proposition 3. *(Justification, Attenuation, Confirmation) Let a_i and a_j be events. We have:*

1. *a_i justifies a_j if and only if $\Pi(\neg a_j \wedge a_i) < \Pi(a_i) \leq \Pi(a_j) = \Pi(\neg a_j) = 1$.*
2. *a_i confirms a_j if and only if $\Pi(\neg a_j \wedge a_i) < \Pi(\neg a_j).\Pi(a_i) < \Pi(a_i)$.*
3. *a_i attenuates a_j if and only if $\Pi(\neg a_j).\Pi(a_i) < \Pi(\neg a_j \wedge a_i) < \Pi(a_i)$.*

4.2 Ascribing Causal Relations in Presence of Interventions

In this section, we extend the model for causal ascription to distinguish between observations and interventions. Independence in the possibilistic framework is defined in presence of observations. In this section, we propose to extend this definition to handle interventions by introducing the 'do' operator in the causal model.

Definition 4. *An event a_j is said independent of the intervention on A_i, denoted $do(a_i)$, if and only if $N(a_j|do(a_i)) = N(a_j)$. Namely, if the intervention $do(a_i)$ does not change our beliefs on a_j.*

Proposition 4. *a_j is independent of the intervention $do(a_i)$ if and only if one of the following properties holds:*

1. *$\exists u_{A_i}$ such that $\Pi(\neg a_j|a_i, u_{A_i}) = \Pi(u_{A_i}) = \Pi(\neg a_j) = 1$*
2. *$\forall u_{A_i}, \Pi(\neg a_j|a_i, u_{A_i}).\Pi(u_{A_i}) = \Pi(\neg a_j) = 0$*
3. *$\max_{u_{A_i}} \Pi(\neg a_j|a_i, u_{A_i}).\Pi(u_{A_i}) = \Pi(\neg a_j)$*

Definition 5. *Assume that we have a sequence of events where an event a_j is rejected and becomes accepted after forcing the variable (or set of variables) A_i to take the value a_i ($do(a_i)$). We say that $do(a_i)$ is the cause of a_j. Formally, an intervention $do(a_i)$ **causes** an event a_j if $N(\neg a_j) > 0$ and $N(a_j|do(a_i)) > 0$.*

Example 5. *Let us consider the same background knowledge given in the example 1. In the context of $A_{back} = 1$, we have $\Pi(ACC = 1 \wedge A_{back} = 1) = 0.6 < \Pi(ACC = 0 \wedge A_{back} = 1) = 1$. Forcing the driver B to brake ($do(B_b = 1)$) rejects the event $ACC = 0$:$\Pi(ACC = 1 \wedge A_{back} = 1 \wedge do(B_b = 1)) = 1 > \Pi(ACC = 0 \wedge A_{back} = 1 \wedge do(B_b = 1)) = 0.4$. The agent judges that $do(B_b = 1)$ is the cause of $ACC = 1$ in the context A_{back}.*

Definition 6. *An intervention $do(a_i)$ **facilitates** the occurrence of an event a_j if $N(\neg a_j) > 0$ and $N(a_j|do(a_i)) = N(\neg a_j|do(a_i)) = 0$.*

Example 6. *Initially, we have $\Pi(ACC = 1) = 0.6 < \Pi(ACC = 0) = 1$. Applying the intervention $do(B_b = 1)$), the event ACC becomes ignored:$\Pi(ACC = 1 \wedge do(B_b = 1)) = \Pi(ACC = 0 \wedge do(B_b = 1)) = 1$. Then, $do(B_b = 1)$ facilitates the occurrence of ACC.*

Definition 7. *An intervention $do(a_i)$ **justifies** the occurrence of an event a_j if $N(a_j) = N(\neg a_j) = 0$ and $N(a_j|do(a_i)) > 0$.*

Attenuation and confirmation in presence of interventions can be defined similarly

Example 7. *Some causal relations disappear with the 'do' operator. For example, initially,* $A_{st} = 1$ *is rejected:* $\Pi(A_{st} = 1) = 0.24 < \Pi(A_{st} = 0) = 1$. *Observing* $A_{CL} = 1$ *attenuates the acceptation of* $A_{st} = 0$*:* $\Pi(A_{st} = 1 \wedge A_{CL} = 1) = 0.4 < \Pi(A_{st} = 0 \wedge A_{CL} = 1) = 1$. *In the meantime, forcing* A *to change the lane* $(do(A_{CL} = 1))$ *does not alter our beliefs on* A_{st}*:* $\Pi(A_{st} = 1 \wedge do(A_{CL} = 1)) = 0.24 < \Pi(A_{st} = 0 \wedge do(A_{CL} = 1)) = 1$.

The following proposition presents some properties of these causal relations in presence of interventions in a quantitative possibilistic frameworks.

Proposition 5. *Let* a_i *and* a_j *be events. Let* u_{A_i} *an instance of the parents of* A_i*. We have:*

1. $do(a_i)$ *causes* a_j *if and only if* $\Pi(a_j) < 1$ *and* $/\exists u_{A_i}, \Pi(\neg a_j, a_i, u_{A_i}) = \Pi(a_i|u_{A_i})$.
2. $do(a_i)$ *facilitates the occurrence of* a_j *if and only if* $\Pi(a_j) < 1$, $\exists u_{A_i}, \Pi(a_j, a_i, u_{A_i}) = \Pi(a_i|u_{A_i})$ *and* $\exists u_{A_i}, \Pi(\neg a_j, a_i, u_{A_i}) = \Pi(a_i|u_{A_i})$.
3. $do(a_i)$ *justifies the occurrence of* a_j *if and only if* $\Pi(\neg a_j) = \Pi(a_j) = 1$ *and* $\forall u_{A_i}, \Pi(\neg a_j|a_i, u_{A_i}).\Pi(u_{A_i}) \neq 1$.
4. $do(a_i)$ *confirms* a_j *if and only if* $\max_{u_{A_i}} \frac{\Pi(\neg a_j, a_i, u_{A_i})}{\Pi(a_i|u_{A_i})} < \Pi(\neg a_j) < \Pi(a_j) = 1$.
5. $do(a_i)$ *attenuates* a_j *if and only if* $\Pi(\neg a_j) < \max_{u_{A_i}} \frac{\Pi(\neg a_j, a_i, u_{A_i})}{\Pi(a_i|u_{A_i})} < \Pi(a_j) = 1$.

4.3 Computational Issues

The main advantage of using quantitative possibilistic causal networks instead of System P to encode background knowledge concerns the computation of causality ascriptions and other influence relations. Indeed, to compute causality (resp. facilitation, justification, attenuation, confirmation) it is enough to have algorithm that computes the basic ingredients $N(a_j|a_i)$ and $N(a_j|do(a_i))$. $N(a_j|a_i)$ is equivalent to $1 - \Pi(\neg a_j|a_i) = 1 - \frac{\Pi(\neg a_j \wedge a_i)}{\Pi(a_i)}$ where $\Pi(\neg a_j \wedge a_i)$ and $\Pi(a_i)$ can be easily computed using propagation algorithms (such as an adaptation of the junction tree algorithm in possibilistic framework) in possibilistic networks. $N(a_j|do(a_i))$ is equivalent to $1 - \Pi(\neg a_j|do(a_i))$. Recall that the effect of an intervention $do(A_i = a_i)$ on a possibility distribution is given, $\forall \omega$, by:

$$\pi(\omega|do(A_i = a_i)) \begin{cases} \prod_{j \neq i} \Pi(a_j|u_j) & if\ \omega \models a_i \\ 0 & else, \end{cases} \tag{5}$$

where a_j (resp. u_j) corresponds to the value of A_j (resp. U_j the parents set of A_j) consistent with the interpretation ω.

One interesting form of equation 5 allows computing the effect of an intervention $do(a_i)$ on the variable (or the set of variable) A_j (disjoint of $\{U_i \cup A_i\}$). This equation is given by the following [4]:

$$\Pi(a_j|do(a_i)) = \max_{U_{A_i}} \Pi(a_j|a_i, U_{A_i}).\pi(U_{A_i}) = \frac{\Pi(a_j \wedge a_i)}{\Pi(a_i|U_{A_i})} \tag{6}$$

where a_i is an event and an instance of the variable (or the set of variables) A_i. U_{A_i} is the parents set of A_i.

In the possibilistic framework, the effect of an intervention may be computed in different manners in presence of both observations and interventions:

- The first solution consists of generalizing equation 6 to handle both observations and interventions. This solution is not satisfactory since when the parents set of variables concerned by interventions have important sizes, this approach is inappropriate and expensive. Namely, this operation requires $O(|D|^{|u|\times|r|}))$ where $|D|$ denotes the variables domains size, $|u|$ the instances number of parents and r is the number of interventions.
- The second solution consists of mutilating graphs. This solution is also excluded since each occurring intervention entails the reconstruction of the junction tree with is the most expensive operation of the propagation process.
- The third solution is the more satisfactory for general structures. This approach consists of augmenting the graph by adding extra nodes to each variable that may be concerned by interventions. The main advantage is to unify observations and interventions treatments through the propagation process. Extra-nodes mean extra costs for the junction tree construction. It has been shown that extra nodes may be added equivalently after the junction tree construction and during the initialization step [4].

5 Conclusion

In this paper, we focused on the causality and influence notions in the possibilistic frameworks using quantitative possibilistic causal networks. We presented a possibilistic model for ascribing causality and other close relations to this notion in the quantitative possibilistic framework. The first advantage of this model (comparing to the qualitative one based on the System P) is its ability to distinguish between simple observations and interventions and thus by introducing the do operator. The aim of such model is not to determine the real cause but to predict the relation between events that an agent may ascribe by observing dynamics of the system. The second advantage is the possibility of reusing existing propagation algorithms of possibilistic networks to compute causality and other concepts introduced in this paper.

Acknowledgments. This work is supported by ANR Blanc MICRAC project.

References

1. Gebhardt, J., Kruse, R.: Possinfer - a software tool for possibilistic inference. In: Dubois, D., Prade, H., Yager, R. (eds.) Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications, New York (1995)
2. Shenoy, P.P.: Using possibility theory in expert systems. Fuzzy Sets and Systems 52, 129–142 (1992)
3. Bonnefon, J.F., Da Silva Neves, R., Dubois, D., Prade, H.: Background default knowledge and causality ascriptions. In: Proceedings of ECAI 2006, pp. 11–15 (2006)
4. Benferhat, S., Smaoui, S.: Possibilistic causal networks for handling interventions: A new propagation algorithm. In: Proceedings of AAAI 2007, pp. 373–378. AAAI Press, Menlo Park (2007)

5. Goldszmidt, M., Pearl, J.: Rank-based systems: A simple approach to belief revision, belief update, and reasoning about evidence and actions. In: Proceeding of KR 1992, pp. 661–672. Kaufmann, San Mateo (1992)
6. Pearl, J.: Causality: models, reasoning, and inference. Cambridge University Press, New York (2000)
7. Dubois, D., Prade, H.: Possibility theory: Qualitative and quantitative aspects. Handbook of Defeasible Reasoning and Uncertainty Management Systems 1, 169–226 (1998)
8. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmman, San Francisco (1988)
9. Spohn, W.: Ordinal conditional functions: a dynamic theory of epistemic states causation in decision. In: Harper, W., Skyrms, B. (eds.) Belief Changes and Statistics, pp. 105–134 (1988)
10. Halpern, J., Pearl, J.: Causes and explanations: A structural-model approach — part i: Causes. In: Proceedings of UAI 2001, pp. 194–200. Morgan Kaufmann, San Francisco (2001)
11. Benferhat, S., Dubois, D., Prade, H.: Nonmonotonic reasoning, conditional objects and possibility theory. Artificial Intelligence 92, 259–276 (1997)
12. Dubois, D., Farias Del Cerro, L., Herzig, A., Prade, H.: A roadmap of qualitative independence. In: Dubois, D., Prade, H., Klement, E. (eds.) Fuzzy Sets, Logics and Reasoning about Knowledge. Applied Logic series, vol. 15, pp. 325–350. Kluwer Academic Publishers, Dordrecht (1999)

Big Bang Big Crunch Optimization Method Based Fuzzy Model Inversion

Tufan Kumbasar, İbrahim Eksin, Müjde Güzelkaya, and Engin Yeşil

Department of Control Engineering
Istanbul Technical University
Maslak, TR-34469, Istanbul, Turkey
{kumbasar,eksin,gkaya,yesil}@elk.itu.edu.tr

Abstract. The inverse fuzzy model can be used as a controller in an open loop fashion to produce perfect control if there does not exist any disturbance or parameter variation in the system. In this paper, a new fuzzy model inversion technique that is based on an evolutionary search algorithm called Big Bang Big Crunch (BB-BC) optimization is introduced. Even though various fuzzy inversion methods can be found in literature, these methods are only applicable under certain conditions or limitations. On the other hand, there does not exist any limitation or condition for the new methodology presented here. In this new technique, the inverse fuzzy model control signal is generated iteratively as a consequence of an optimization operation. Since the BB-BC optimization algorithm has a high convergence speed and low computational time, the optimal inverse fuzzy model control signal is generated within each sampling time. The beneficial sides of the open loop control approach based on the proposed fuzzy model inversion technique are illustrated through two simulation studies.

1 Introduction

The use of the inverse system model might be an efficient way in control nonlinear system. Even when the necessary and sufficient conditions of obtaining the inverse of a nonlinear system have not been guaranteed, some numerical methods have been proposed which generates the inverse of the nonlinear system [1]. It is a known fact that fuzzy models can represent highly nonlinear processes. Hence, describing a nonlinear system by a fuzzy model may provide one an opportunity to make an effective control on the process using its inverse and there exists various fuzzy inversion methods developed in literature [2].

Lately different fuzzy inversion techniques for certain fuzzy models have been suggested that can only be applied under certain limitations. In [3] it is claimed that the exact inverse of the model can be attained when the system is modelled as singleton fuzzy type. However, it is required that the singleton type fuzzy model must satisfy some invertibility conditions. When these conditions are not satisfied the fuzzy model must be decomposed into invertible parts [4]. Another inversion method has been introduced by in [5]. In this method, a simple nonlinear state observer is used

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 732–740, 2008.

that reconstructs the selected input of the forward fuzzy model from its output using an appropriate strategy and a copy of the fuzzy model itself [6]. This method does not require any invertibility condition to be satisfied. Furthermore, the method proposed in [7], the fuzzy model is decomposed into fuzzy meshes and the inverse of the global fuzzy system is obtained through inversion of each fuzzy mesh. In this inversion method, there is again no need to check any invertibility condition [8]. In some of the other inversion methods, the inverse model is obtained directly mapping the output and input data of the process via neural networks or fuzzy logic approach [9, 10].

Recently, a new evolutionary computation algorithm named as Big Bang Big Crunch (BB-BC) is presented in [11]. The leading advantage of BB-BC is the high convergence speed and the low computation time. The working principle of this method can be explained as the transformation of a convergent solution to a chaotic state and then back to a single tentative solution point. This evolutionary search algorithm was first used for on line adaptation of the fuzzy model by updating the consequent parameters of the model [12].

In this paper, a fuzzy model based open loop control structure is proposed in which the inverse fuzzy model is used as the controller. In this new setting, the output of the inverse fuzzy model which would be the control signal for the system would be generated via an optimization problem. The optimization problem can be defined as to decrease the error between the fuzzy model output and the reference signal. As the fuzzy model output converges to the set point; the process output will converge to the set point; unless there does not exist any disturbance or parameter variation in the system. Since the BB-BC optimization algorithm has a high convergence speed and low computational time, the optimal inverse fuzzy model control signal is generated within each sampling time.

In the next chapter, brief information about the BB-BC optimization technique is given. In the third chapter, the BB-BC optimization based inverse fuzzy model controller is introduced. Finally, the performance of this control structure is illustrated on two simulation studies.

2 Big Bang–Big Crunch Optimization

The Big Bang-Big Crunch (BB-BC) optimization method is built on two main steps: The first step is the Big Bang phase where candidate solutions are randomly distributed over the search space and the next step is the Big Crunch where a contraction procedure calculates a center of mass for the population. The initial Big Bang population is randomly generated over the entire search space just like the other evolutionary search algorithms. All subsequent Big Bang phases are randomly distributed about the center of mass or the best fit individual in a similar fashion. In [11], the working principle of this evolutionary method is explained as to transform a convergent solution to a chaotic state which is a new set of solutions. The procedure of the BB-BC optimization is given in the table below:

Table 1. BB-BC Optimization Algorithm

Step 1 *(Big Bang Phase)* An initial generation of N candidates is generated randomly in the search space. **Step 2** The cost function values of all the candidate solutions are computed. **Step 3** *(Big Crunch Phase)* The center of mass is calculated. Either the best fit individual or the center of mass is chosen as the point of Big Bang Phase. **Step 4** New candidates are calculated around the new point calculated in Step 3 by adding or subtracting a random number whose value decreases as the iterations elapse. **Step 5** Return to Step 2 until stopping criteria has been met.

After the Big Bang, a contraction procedure is applied during the Big Crunch. In this phase, the contraction operator takes the current positions of each candidate solution in the population and its associated cost function value and computes a center of mass. The center of mass can be computed as:

$$x_c = \frac{\sum_{i=1}^{N} \frac{1}{f^i} x_i}{\sum_{i=1}^{N} \frac{1}{f^i}} \tag{1}$$

where x_c = position of the center of mass; x_i = position of candidate; f^i =cost function value of candidate i; and N = population size. Instead of the position of the center of mass, the best fit individual can also be chosen as the starting point in the Big Bang phase.

The new generation for the next iteration Big Bang phase is normally distributed around x_c

$$x_i^{new} = x_c + \sigma \tag{2}$$

where x_i^{new} = the new candidate solution i; and σ standard deviation of a standard normal distribution. The standard deviation decreases as the iterations elapse according to the following formula

$$\sigma = \frac{r\alpha(x_{\max} - x_{\min})}{k} \tag{3}$$

where r is random number; α is a parameter limiting the size of the search space, $x_{\max}$ and $x_{\min}$ are the upper and lower limits; and k is the number of the iterations. Therefore, the new point is generated as follows:

$$x_i^{new} = x_c + \frac{r\alpha(x_{\max} - x_{\min})}{k} \tag{4}$$

Since normally distributed numbers can be exceeding ±1, it is necessary to limit population to the prescribed search space boundaries. This narrowing down restricts the candidate solutions into the search space boundaries [11].

3 BB-BC based Inverse Fuzzy Controller

In this proposed control structure, the control signal generation is handled as an optimization problem. The problem can be defined as to decrease the error between the fuzzy model output and the reference signal. Assuming that the fuzzy model matches the process perfectly, as the fuzzy model output converges to the set point; the process output will converge to the set point, too. Since an online implementation of this evolutionary algorithm is feasible, at each sampling time the optimal control signal can be generated. Then the optimal control signal is then applied to the process.

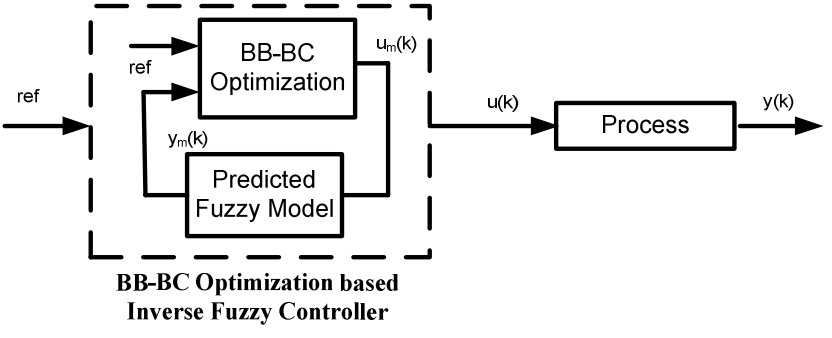


Fig. 1. BB-BC based Inverse Fuzzy Control Scheme

The cost function J_1 which is minimized at every sampling time is chosen as:

$$J_1 = \left| ref - \mathrm{y_m}(k) \right| \tag{5}$$

The reason why a predicted fuzzy model is used is to eliminate effect of the time delay. If the model has no time delay, the fuzzy model can be used directly for optimization.

Simulation Study1: First Order Process with Time Delay

In the first simulation study, the proposed control structure has been tested on a first order process with time delay. The process model is given as:

$$G(s) = \frac{0.97}{0.52s + 1} e^{-0.2s} \tag{6}$$

First a fuzzy model of the process is obtained. For this purpose an identification experiment is performed. In this identification experiment the sampling time is selected to be 0.1s. Therefore, the input of the fuzzy model is chosen as u(k-3) and y(k-1) and the collected output and input data has then been trained by ANFIS toolbox to obtain the singleton fuzzy model. The membership functions of the fuzzy model are defined with four triangular functions.

The boundaries of the control signal have been chosen as the follows:

$$U_{max}=1,\ U_{min}=0$$

The parameters for the evolutionary algorithm have been chosen as:
Population size=20, Iteration number=20

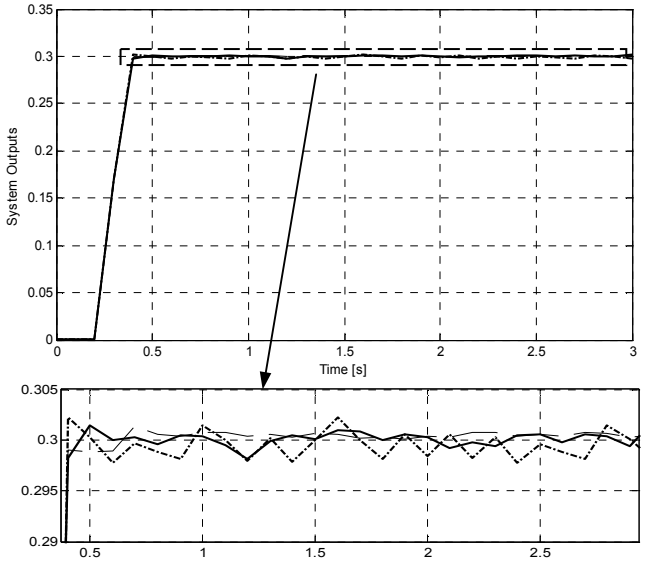


Fig. 2. Three system outputs for the different trials of the BB-BC optimization algorithm for the reference signal 0.3

Since BB-BC is a stochastic evolutionary algorithm, the performance of the controller will vary for each trial. Therefore, three system outputs for the different trials of the BB-BC optimization algorithm have been shown in the Fig.2.

Simulation Study 2: Nonlinear Level Control System

In the second simulation study, level control of a spherical tank is considered [18]. A schematic diagram of the tank level system is shown in Fig.4.

The differential equation related to the tank level system is given as:

$$Q_i(t-d)-Q_o=\pi R^2\left[1-\frac{(R^2-y)}{R^2}\right]\frac{dy}{dt} \tag{7}$$

where R is the radius of the spherical tank, Q_i is the inlet volumetric flow rate, and Q_o is the outlet flow rate. The delay from the manipulated input Q_i to the controlled

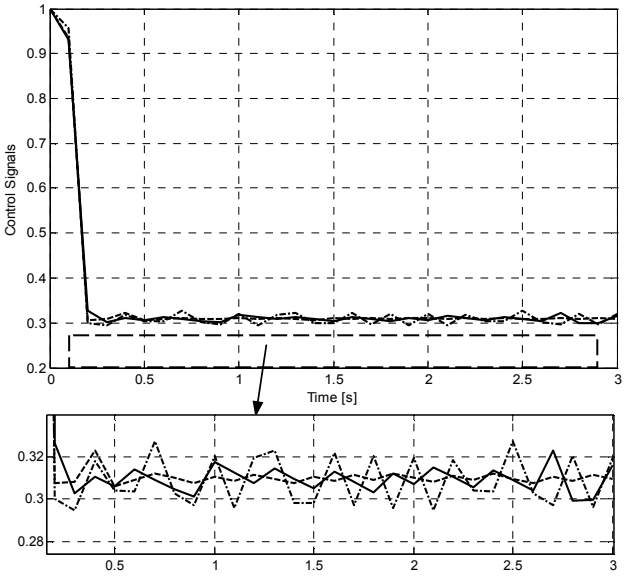


Fig. 3. The three control signals for the different trials of the BB-BC optimization algorithm for the reference signal 0.3

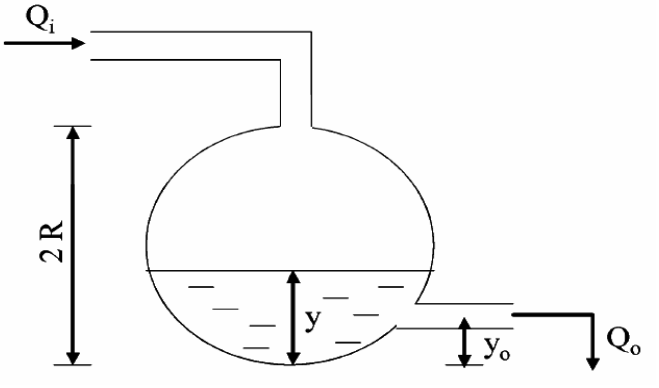


Fig. 4. Spherical tank

output y is indicated by d. The outlet flow rate Q_o is related to the level y via the Bernoulli equation:

$$Q_o = \sqrt{2g(y - y_0)} \tag{8}$$

where g represents the gravitation constant and y_o represents the height of the outlet pipe as measured from the base of the column. The parameters of the process are assumed to be R=1 m, d =0s, and y_o=0.1 m in this simulation. The nominal operating value of y is taken as 0.5 m. At this nominal operating value, the output y will respond faster than the case when y is close to 1. Just like in the previous simulation

study, the Takagi-Sugeno fuzzy model of the process is obtained with the help of ANFIS toolbox (T_s=0.05s) and the regression vector has been chosen as:

$$x=[y(k-1), u(k-1)] \tag{9}$$

In order to represent the nonlinearities of spherical tank better, the membership functions of the fuzzy model are defined with 8 gauss functions. It is assumed that the tank is working at the operating level 0.5m. In the first part, the reference is changed from 0.5m to 1.2m.

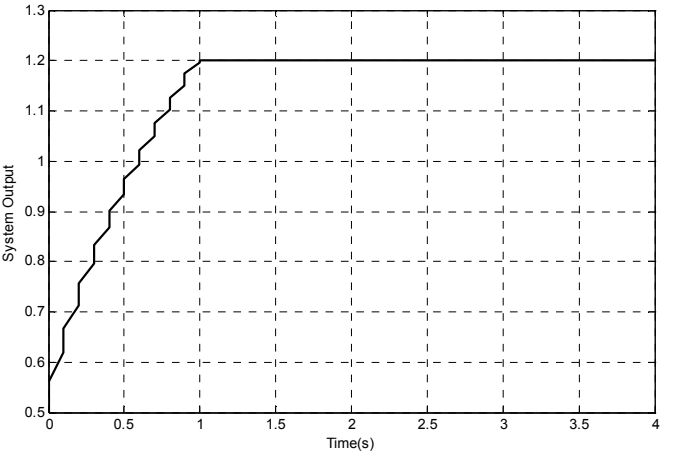


Fig. 5. The system output for the reference of 1.2 m

The boundaries of the control signal for this simulation study have been chosen as the follows:

$$U_{max}=6,\ U_{min}=0$$

The parameters for the evolutionary algorithm have been chosen as:

Population size=20, Iteration number=20

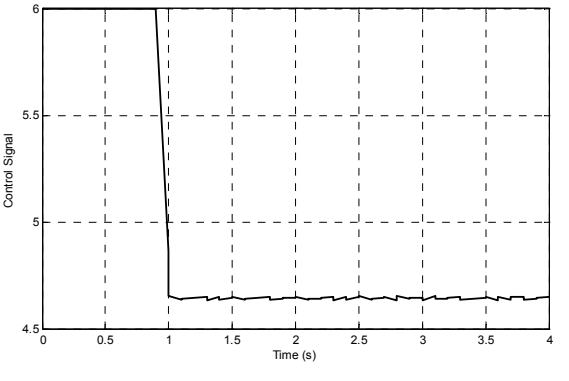
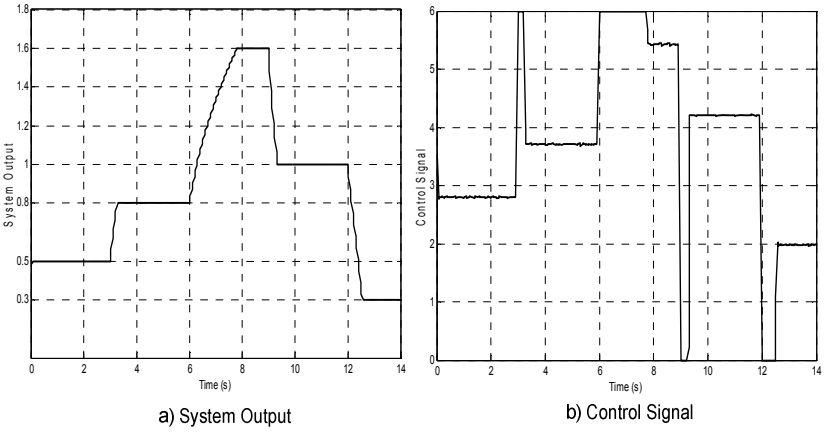


Fig. 6. The generated inverse fuzzy model control signal for the reference signal of 1.2 m

In the second part, since the nonlinearity is related to the level of the tank, the controller has been tested under varying reference values (Fig.7). It can be seen from Fig.7 that the controller provides satisfactory performances for different reference signals.



a) System Output b) Control Signal

Fig. 7. The system output and the control signal for varying reference values

4 Conclusions

In this study, a new iterative inversion technique based on the BB-BC optimization has been presented. The main advantage of this inversion method is that unlike the other present inversion techniques it does not require any conditions to be satisfied. In this new technique, the inverse fuzzy model control signal generation is handled as an optimization problem. Since the BB-BC optimization algorithm has a high convergence speed and low computational time, the optimal inverse fuzzy model control signal can be generated within each sampling time in an on line fashion. In order to show performance of the new approach two simulation studies have been performed. It has been shown that both the linear model with time delay and nonlinear model could have been controlled perfectly in an open loop control scheme. Since the proposed based on a stochastic global search algorithm, the generated inverse fuzzy model control signals will vary at each trial which is the only arguable point of this inversion technique. However, as it has been demonstrated in simulation examples given in this study, all trials have provided quite satisfactory results.

References

1. Economou, G., Morari, M., Palsson, B.: Internal model control. Extension to nonlinear systems. Ind. Eng. Chem. Process Des. Dev. 25, 403–411 (1986)
2. Babuska, R.: Fuzzy Modeling for Control. Kluwer Academic Publishers, Boston (1998)

3. Babuska, R., Sousa, J., Verbruggen, H.B.: Model Based Design of Fuzzy Control Systems. In: Proc. of the 3rd Europ. Congr. on 6rt. Techniques and Soft Computing EUFlT 1995, Aachen, Germany, pp. 837–841 (1995)
4. Baranyi, P., Korondi, P., Hashimoto, H., Wada, M.: Fuzzy Inversion and Rule Base Reduction. In: IEEE Int. Conf. on Intelligent Engineering Systems (IEEE INES 1997), Budapest, pp. 301–306 (1997)
5. Varkonyi-Koczy, A.R., Amos, A., Kovicshizy, T.: Genetic Algorithms in Fuzzy Model Inversion. In: IEEE International Fuzzy Systems Conference Proceedings, Seoul,Korea, August 22-25 (1999)
6. Varkonyi-Koczy, A.R., Péceli, G., Dobrowiecki, T.P., Kovácsházy, T.: Iterative Fuzzy Model Inversion. In: Proc. of the 1998 IEEE Int. Conference on Fuzzy Systems, FUZZ-IEEE 1998, Anchorage, Alaska, USA, May 5-9, vol. 1, pp. 561–566 (1998)
7. Boukezzoula, R., Galichet, S., Folloy, L.: Nonlinear Internal Model Control: Applicaation of Inverse Model Based Fuzzy Control. IEEE Transactions on Fuzzy Systems 11(6) (2003)
8. Boukezzoula, R., Galichet, S., Folloy, L.: Fuzzy Feedback Linearizing Controller and its Equivalence with the Fuzzy Nonlinear Internal Model Control Structure. International Journal of Applied Mathematics and Computer Science 17(2), 233–248 (2007)
9. Abonyi, J., Andersen, H., Nagy, L., Szeifert, F.: Inverse Fuzzy-Process –Model Based Adaptive Control. Mathematics and Computers in Simulation 51(1-2), 119–132 (1999)
10. Fischer, M., Isermann, R.: Robust hybrid control based on inverse fuzzy process models. In: IEEE International conference on fuzzy systems, New Orleans, USA, pp. 1210–1216 (1996)
11. Erol, O.K., Eksin, I.: A new optimization method: Big Bang–Big Crunch. Advances in Engineering Software 37, 106–111 (2006)
12. Kumbasar, T., Yeşil, E., Eksin, İ., Güzelkaya, M.: Inverse Fuzzy Model Control with Online Adaptation via Big Bang-Big Crunch Optimization. In: The 3rd International Symposium on Communications, Control and Signal Processing, ISCCSP, Malta (2008)
13. Agrawal, P., Lakshminarayan, S.: Tuning proportional-integral-derivative controllers using achievable performance indices. Ind. Eng. Chem. Res. 42, 5576–5582 (2003)

Choquet Integrals and OWA Criteria as a Natural (and Optimal) Next Step after Linear Aggregation: A New General Justification

François Modave, Martine Ceberio, and Vladik Kreinovich

Department of Computer Science
University of Texas at El Paso, El Paso, TX 79968, USA
{fmodave,mceberio,vladik}@utep.edu

Abstract. In multi-criteria decision making, it is necessary to aggregate (combine) utility values corresponding to several criteria (parameters). The simplest way to combine these values is to use linear aggregation. In many practical situations, however, linear aggregation does not fully adequately describe the actual decision making process, so non-linear aggregation is needed.

From the purely mathematical viewpoint, the next natural step after linear functions is the use of quadratic functions. However, in decision making, a different type of non-linearities are usually more adequate than quadratic ones: non-linearities like OWA or Choquet integral that use min and max in addition to linear combinations. In this paper, we explain the empirically observed advantage of such aggregation operations.

1 Introduction

One of the main purposes of Artificial Intelligence in general is to incorporate a large part of human intelligent reasoning and decision-making into a computer-based systems, so that the resulting intelligent computer-based systems help users in making rational decisions. In particular, to help a user make a decision among a large number of alternatives, an intelligent decision-making systems should select a small number of these alternatives – alternatives which are of the most potential interest to the user.

For example, with so many possible houses on the market, it is not realistically possible to have a potential buyer inspect all the houses sold in a given city. Instead, a good realtor tries to find out the buyer's preferences and only show him or her houses that more or less fit these preferences. It would be great to have an automated system for making similar pre-selections.

To be able to make this selection, we must elicit the information about the user preferences.

In principle, we can get a full picture of the user preferences by asking the user to compare and/or rank all possible alternatives. Such a complete description of user preferences may be sometimes useful, but in decision making applications, such an extensive question-asking defeats the whole purpose of intelligent

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 741–753, 2008.

decision-making systems – to avoid requiring that the the user make a large number of comparisons.

The existing approach to this problem is called *multi-criteria decision making* (MCDM). The main idea behind this approach is that each alternative is characterized by the values of different parameters. For example, the buyer's selection of a house depends on the house's size, on its age, on its geographical location, on the number of bedrooms and bathrooms, etc. The idea is to elicit preferences corresponding to each of these parameters, and then to combine these single-parameter preferences into a reasonable model for describing the user's choice.

In the standard decision making theory, preferences are characterized by assigning, to each alternative, a numerical value called its *utility*. In these terms, the multi-criteria decision making approach means that we try to combine single-variable utility values $u_1(x_1), \ldots, u_n(x_n)$ characterizing the user's preferences over individual parameters $x_1, \ldots, x_n$ into a utility value $u(x_1, \ldots, x_n)$ that characterizes the utility of an alternative described by the values $(x_1, \ldots, x_n)$.

In the first approximation, it makes sense simply to add the individual utility values with appropriate weights, i.e., to consider linear aggregation

$$u(x_1, \ldots, x_n) = w_1 \cdot u_1(x_1) + \ldots + w_n \cdot u_n(x_n).$$

In many practical situations, linear aggregation works well, but in some cases, it leads to counterintuitive conclusions. For example, when selecting a house, a user can assign certain weights to all the parameters characterizing different houses, but the user may also has absolute limitations: e.g., a user with kids may want a house with at least two bedrooms, and no advantages in location and price would entice her to buy a one-bedroom house. To describe such reasonable preferences, we must therefore go beyond linear aggregation functions.

From the purely mathematical viewpoint, the inadequacy of a linear model is a particular example of a very typical situation. Often, when we describe the actual dependence between the quantities in physics, chemistry, engineering, etc., a linear expressions $y = c_0 + c_1 \cdot x_1 + \ldots + c_n \cdot x_n$ is a very good first approximation (at least locally), but to get a more accurate approximations, we must take non-linearity into account. In mathematical applications to physics, engineering, etc., there is a standard way to take non-linearity into account: if a linear approximation is not accurate enough, a natural idea is to use a quadratic approximation $y \approx a_0 + \sum\limits_{i=1}^{n} c_i \cdot x_i + \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} c_{ij} \cdot x_i \cdot x_j$; if the quadratic approximation is not sufficient accurate, we can use a cubic approximation, etc.; see, e.g., [3].

At first glance, it seems reasonable to apply a similar idea to multi-criteria decision making and consider quadratic aggregation functions

$$u \stackrel{\text{def}}{=} u(x_1, \ldots, x_n) = u_0 + \sum_{i=1}^{n} w_i \cdot u_i(x_i) + \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \cdot u_i(x_i) \cdot u_j(x_j).$$

Surprisingly, in contrast to physics and engineering applications, quadratic approximation do not work as well as approximations based on the use of piece-wise

linear functions, such as the OWA operation $u = w_1 \cdot u_{(1)} + \ldots + w_n \cdot u_{(n)}$, where $u_{(1)} = \max(u_1(x_1), \ldots, u_n(x_n))$ is the largest of n utility values $u_i(x_i)$, $u_{(2)}$ is the second largest, $\ldots$, and $u_{(n)} = \min(u_1(x_1), \ldots, u_n(x_n))$ is the smallest of n utility values; see, e.g., [9].

In our own research, we have applied OWA and we have also applied similar piece-wise linear operations (based on the so-called Choquet integral [4]), and we also got good results – better than quadratic approximations; see, e.g., [1] and references therein. Similar results have been obtained by others. For quite some time, why piece-wise approximations are better than quadratic ones remains a mystery to us – and to many other researchers whom we asked this question. Now, we finally have an answer to this question – and this answer is presented in the current paper.

Thus, the paper provides a new justification of the use of piece-wise aggregation operations in multi-criteria decision making – a justification that explains why these aggregation operations are better than the (seemingly more natural) quadratic ones.

The structure of this paper is as follows. To explain our answer to the long-standing puzzle, we need to recall the properties of the utility functions. The needed properties of utility functions are described in Section 2. Readers who are already well familiar with the standard decision making theory (and with the corresponding properties of utility functions) can skip this section and proceed directly to Section 3. In Section 3, we explain why quadratic aggregation operations are less adequate than OWA and Choquet operations: because the basic properties of utility functions lead to the need for aggregation operations to be scale-invariant; OWA and Choquet aggregations are scale-invariant, while quadratic aggregations aren't.

In Section 4, we explain that OWA and Choquet operations are, in some reasonable sense, the most general ones: namely, crudely speaking, every scale-invariant operation can be composed of linear combinations and min and max operations. We also argue that the selection of linear operations, min, and max as elementary operations is well justified from the computational viewpoint: since they are the fastest possible scale-invariant operations. This justification is presented in Section 5. The mathematical proofs are placed, for reader's convenience, in a special Section 6. The last section contains conclusions.

2 Standard Decision Making Theory: A Brief Reminder

As we have mentioned earlier, to explain our answer to the long-standing puzzle, we need to recall the properties of the utility functions. The needed properties of utility functions are described in this section. Readers who are already well familiar with the standard decision making theory (and with the corresponding properties of utility functions) can skip this section and proceed directly to Section 3.

To be able to describe decisions, we must have a numerical scale for describing preferences. The traditional decision making theory (see, e.g., [5,6,7]) starts with

an observation that such a scale can be naturally obtained by using probabilities. Specifically, to design this scale, we select two alternatives:

- a very negative alternative A_0; e.g., an alternative in which the decision maker loses all his money (and/or loses his health as well), and
- a very positive alternative A_1; e.g., an alternative in which the decision maker wins several million dollars.

Based on these two alternatives, we can, for every value $p \in [0,1]$, consider a randomized alternative $L(p)$ in which we get A_1 with probability p and A_0 with probability $1-p$.

(It should be mentioned that in the standard decision making theory, randomized alternatives like $L(p)$ are also (somewhat misleadingly) called *lotteries*. This name comes from the fact that a lottery is one of the few real-life examples of randomized outcomes with known probabilities.)

In the two extreme cases $p=0$ and $p=1$, the randomized alternative $L(p)$ turns into one of the original alternatives: when $p=1$, we get the favorable alternative A_1 (with probability 1), and when $p=0$, we get the unfavorable alternative A_0. In general, the larger the probability p of the favorable alternative A_1, the more preferable is the corresponding randomized alternative $L(p)$. Thus, the corresponding randomized alternatives ("lotteries") $L(p)$ form a continuous 1-D scale ranging from the very negative alternative A_0 to the very positive alternative A_1.

So, it is reasonable to gauge the preference of an arbitrary alternative A by comparing it to different alternatives $L(p)$ from this scale until we find A's place on this scale, i.e., the value $p \in [0,1]$ for which, to this decision maker, the alternative A is equivalent to $L(p)$: $L(p) \sim A$. This value is called the *utility* $u(A)$ of the alternative A in the standard decision making theory.

In our definition, the numerical value of the utility depends on the selection of the alternatives A_0 and A_1: e.g., A_0 is the alternative whose utility is 0 and A_1 is the alternative whose utility is 1. What if we use a different set of alternatives, e.g., $A_0' < A_0$ and $A_1' > A_1$?

Let A be an arbitrary alternative between A_0 and A_1, and let $u(A)$ be its utility with respect to A_0 and A_1. In other words, we assume that A is equivalent to the randomized alternative in which:

- we have A_1 with probability $u(A)$, and
- we have A_0 with probability $1-p$.

In the scale defined by the new alternatives A_0' and A_1', let $u'(A_0)$, $u'(A_1)$, and $u'(A)$ denote the utilities of A_0, A_1, and A. This means, in particular:

- that A_0 is equivalent to the randomized alternative in which we get A_1' with probability $u'(A_0)$ and A_0' with probability $1-u'(A_0)$; and
- that A_1 is equivalent to the randomized alternative in which we get A_1' with probability $u'(A_1)$ and A_0' with probability $1-u'(A_1)$.

Thus, the alternative A is equivalent to the compound randomized alternative, in which

– first, we select A_1 or A_0 with probabilities $u(A)$ and $1 - u(A)$, and then
– depending on the first selection, we select A'_1 with probability $u'(A_1)$ or $u'(A_0)$ – and A'_0 with the remaining probability.

As the result of this two-stage process, we get either A'_0 or A'_1. The probability p of getting A'_1 in this two-stage process can be computed by using the formula of full probability

$$p = u(A) \cdot u'(A_1) + (1 - u(A)) \cdot u'(A_0) = u(A) \cdot (u'(A_1) - u'(A_0)) + u'(A_0).$$

So, the alternative A is equivalent to a randomized alternative in which we get A'_1 with probability p and A'_0 with the remaining probability $1 - p$. By definition of utility, this means that the utility $u'(A)$ of the alternative A in the scale defined by A'_0 and A'_1 is equal to this value p:

$$u'(A) = u(A) \cdot (u'(A_1) - u'(A_0)) + u'(A_0).$$

So, changing the scale means a linear re-scaling of the utility values:

$$u(A) \to u'(A) = \lambda \cdot u(A) + b$$

for $\lambda = u'(A_1) - u'(A_0) > 0$ and $b = u'(A_0)$.

Vice versa, for every $\lambda > 0$ and b, one can find appropriate events A'_0 and A'_1 for which the re-scaling has exactly these values λ and b. In other words, utility is defined modulo an arbitrary (increasing) linear transformation.

The last important aspect of the standard decision making theory is its description of the results of different actions. Suppose that an action leads to alternatives $a_1, \ldots, a_m$ with probabilities $p_1, \ldots, p_m$. We can assume that we have already determined the utility $u_i = u(a_i)$ of each of the alternatives $a_1, \ldots, a_m$. By definition of the utility, this means that for each i, the alternative a_i is equivalent to the randomized alternative $L(u_i)$ in which we get A_1 with probability u_i and A_0 with probability $1 - u_i$. Thus, the results of the action are equivalent to the two-stage process in which, with the probability p_i, we select a randomized alternative $L(u_i)$. In this two-stage process, the results are either A_1 or A_0. The probability p of getting A_1 in this two-stage process can be computed by using the formula for full probability: $p = p_1 \cdot u_1 + \ldots + p_m \cdot u_m$. Thus, the action is equivalent to a randomized alternative in which we get A_1 with probability p and A_0 with the remaining probability $1 - p$. By definition of utility, this means that the utility u of the action in question is equal to

$$u = p_1 \cdot u_1 + \ldots + p_m \cdot u_m.$$

In statistics, the right-hand of this formula is known as the *expected value*. Thus, we can conclude that the utility of each action with different possible alternatives is equal to the expected value of the utility.

3 Why Quadratic Aggregation Operations Are Less Adequate Than OWA and Choquet Operations: An Explanation

To adequately describe the decision maker's preferences, we must be able, given an alternative characterized by n parameters $x_1, \ldots, x_n$, to describe the utility $u(x_1, \ldots, x_n)$ of this alternative. To get a perfect description of the user's preference, we must elicit such a utility value for all possible combinations of parameters. As we have mentioned in the Introduction, for practical values n, it is not realistic to elicit that many utility values from a user. So, instead, we elicit the user's preference over each of the parameters x_i, and then aggregate the resulting utility values $u_i(x_i)$ into an approximation for $u(x_1, \ldots, x_n)$: $u(x_1, \ldots, x_n) \approx f(u_1, \ldots, u_n)$, where $u_i \stackrel{\text{def}}{=} u_i(x_i)$.

We have also mentioned that in the first approximation, linear aggregation operations $f(u_1, \ldots, u_n) = a_0 + \sum_{i=1}^{n} w_i \cdot u_i$ work well, but to get a more adequate representation of the user's preferences, we must go beyond linear functions. From the purely mathematical viewpoint, it may seem that quadratic functions $f(u_1, \ldots, u_n)$ should provide a reasonable next approximation, but in practice, piece-wise linear aggregation operations such as OWA (or Choquet integral) provide a much more adequate description of expert preferences.

For example, for two parameters, the general OWA combination of two utility values has the form

$$f(u_1, u_2) = w_1 \cdot \min(u_1, u_2) + w_2 \cdot \max(u_1, u_2).$$

Similarly, the general OWA combination of three utility values has the form

$$f(u_1, u_2, u_3) = w_1 \cdot \min(u_1, u_2, u_3) +$$

$$w_2 \cdot \max(\min(u_1, u_2), \min(u_1, u_3), \min(u_2, u_3)) + w_3 \cdot \max(u_1, u_2, u_3).$$

Let us show that this seemingly mysterious advantage of non-quadratic aggregation operations can be explained based on the main properties of the utility functions.

Indeed, as we have mentioned in Section 2, the utility is defined modulo *two* types of transformations: changing a starting point $u \to u + b$ and changing a scale $u \to \lambda \cdot u$ for some $\lambda > 0$. It is therefore reasonable to require that the aggregation operation should not depend on which "unit" (i.e., which extreme event A_1) we use to describe utility. Let us describe this requirement in precise terms.

In the original scale,

- we start with utility values $u_1, \ldots, u_n$;
- to these values, we apply the aggregation operation $f(u_1, \ldots, u_n)$ and get the resulting overall utility $u = f(u_1, \ldots, u_n)$.

On the other hand,

- we can express the same utility values in a new scale, as $u'_1 = \lambda \cdot u_1, \ldots, u'_n = \lambda \cdot u_n$;
- then, we use the same aggregation function to combine the new utility values; as a result, we get the resulting overall utility $u' = f(u'_1, \ldots, u'_n)$.

Substituting the expressions $u'_i = \lambda \cdot u_i$ into this formula, we conclude that $u' = f(\lambda \cdot u_1, \ldots, \lambda \cdot u_n)$. We require that the utility

$$u' = f(u'_1, \ldots, u'_n) = f(\lambda \cdot u_1, \ldots, \lambda \cdot u_n)$$

reflect the same degree of preference as the utility $u = f(u_1, \ldots, u_n)$ but in a different scale: $u' = \lambda \cdot u$, i.e.,

$$f(\lambda \cdot u_1, \ldots, \lambda \cdot u_n) = \lambda \cdot f(u_1, \ldots, u_n).$$

It is worth mentioning that in mathematics, such functions are called *homogeneous* (of first degree). So, we arrive at the conclusion that an adequate aggregation operation should be homogeneous.

This conclusion about the above mysterious fact. On the other hand, one can show that linear aggregation operations and piece-wise linear aggregation operations like OWA are scale-invariant.

Let us start with a linear aggregation operation $f(u_1, \ldots, u_n) = w_1 \cdot u_1 + \ldots + w_n \cdot u_n$. For this operation, we get

$$f(\lambda \cdot u_1, \ldots, \lambda \cdot u_n) = w_1 \cdot (\lambda \cdot u_1) + \ldots + w_n \cdot (\lambda \cdot u_n) =$$

$$\lambda \cdot (w_1 \cdot u_1 + \ldots + w_n \cdot u_n) = \lambda \cdot f(u_1, \ldots, u_n).$$

Let us now consider the OWA aggregation operation $f(u_1, \ldots, u_n) = w_1 \cdot u_{(1)} + \ldots + w_n \cdot u_{(n)}$, where $u_{(1)}$ is the largest of n values $u_1, \ldots, u_n$, $u_{(2)}$ is the second largest, etc. If we multiply all the utility values u_i by the same constant $\lambda > 0$, their order does not change. In particular, this means that the same value $u_{(1)}$ which was the largest in the original scale is the largest in the new scale as well. Thus, its numerical value $u'_{(1)}$ can be obtained by re-scaling $u_{(1)}$: $u'_{(1)} = \lambda \cdot u_{(1)}$. Similarly, the same value $u_{(2)}$ which was the second largest in the original scale is the second largest in the new scale as well. Thus, its numerical value $u'_{(2)}$ can be obtained by re-scaling $u_{(2)}$: $u'_{(2)} = \lambda \cdot u_{(2)}$, etc. So, we have $u'_{(i)} = \lambda \cdot u_{(i)}$ for all i. Thus, for the OWA aggregation operation, we have

$$f(\lambda \cdot u_1, \ldots, \lambda \cdot u_n) = w_1 \cdot u'_{(1)} + \ldots + w_n \cdot u'_{(n)} = w_1 \cdot (\lambda \cdot u_{(1)}) + \ldots + w_n \cdot (\lambda \cdot u_{(n)}) =$$

$$\lambda \cdot (w_1 \cdot u_{(1)} + \ldots + w_n \cdot u_{(n)}) = \lambda \cdot f(u_1, \ldots, u_n).$$

On the other hand, a generic quadratic operation is not homogeneous. Indeed, a general quadratic operation has the form

$$f(u_1, \ldots, u_n) = \sum_{i=1}^{n} w_i \cdot u_i + \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \cdot u_i \cdot u_j.$$

Here,

$$f(\lambda u_1, \ldots, \lambda u_n) = \sum_{i=1}^{n} w_i \cdot (\lambda \cdot u_i) + \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \cdot (\lambda \cdot u_i) \cdot (\lambda \cdot u_j) =$$

$$\lambda \cdot \sum_{i=1}^{n} w_i \cdot u_i + \lambda^2 \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \cdot u_i \cdot u_j.$$

On the other hand,

$$\lambda \cdot f(u_1, \ldots, u_n) = \lambda \cdot \sum_{i=1}^{n} w_i \cdot u_i + \lambda \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \cdot u_i \cdot u_j.$$

The linear terms in the expressions $f(\lambda u_1, \ldots, \lambda u_n)$ and $\lambda \cdot f(u_1, \ldots, u_n)$ coincide, but the quadratic terms differ: the quadratic term in $f(\lambda u_1, \ldots, \lambda u_n)$ differs from the quadratic term in $\lambda \cdot f(u_1, \ldots, u_n)$ by a factor of λ. Thus, the only possibility to satisfy the scale-invariance (homogeneity) requirement for all λ is to have these differing quadratic terms equal to 0, i.e., to have $w_{ij} = 0$ – but in this case the aggregation operation is linear. So, quadratic operations are indeed not homogeneous – which explains whey they are less adequate in describing user's preferences than homogeneous operations like OWA or Choquet integral.

4 OWA and Choquet Operations Are, in Some Reasonable Sense, the Most General Ones: A New Result

In the previous section, we explained the empirical fact that in multi-criteria decision making, OWA and Choquet operations lead to more adequate results than seemingly natural quadratic aggregation operations. The explanation is that, due to the known properties of the utility, it is reasonable to require that aggregation operation be scale-invariant (homogeneous); OWA and Choquet operations are scale-invariant but quadratic operations are not.

However, in principle, OWA and Choquet operations are just a few examples of scale-invariant operations, so by itself, the above result does not explain why OWA and Choquet operations are so successful and not any other scale-invariant operation. In this section, we give such an explanation.

This explanation is based on the fact that OWA and Choquet operations are compositions of linear functions, min, and max. In this section, we prove that, crudely speaking, every scale-invariant operation can be composed of linear functions and min and max operations.

Definition 1. *A function $f(x_1, \ldots, x_n)$ is called homogeneous if for every x_1, ..., x_n and for every $\lambda > 0$, we have $f(\lambda \cdot x_1, \ldots, \lambda \cdot x_n) = \lambda \cdot f(x_1, \ldots, x_n)$.*

Definition 2. *By a* basic function*, we mean one of the following functions:*

- *a* linear function $f(x_1, \ldots, x_n) = w_1 \cdot x_1 + \ldots + w_n \cdot x_n$*;*
- *a* minimum function $f(x_1, \ldots, x_n) = \min(x_{i_1}, \ldots, x_{i_m})$*; and*
- *a* maximum function $f(x_1, \ldots, x_n) = \max(x_{i_1}, \ldots, x_{i_m})$.

We also say that basic functions are 1-level compositions *of basic functions. We say that a function $f(x_1, \ldots, x_n)$ is a* k-level composition of basic functions *if $f(x_1, \ldots, x_n) = g(h_1(x_1, \ldots, x_n), \ldots, h_m(x_1, \ldots, x_n))$, where g is a basic function, and the functions $h_1(x_1, \ldots, x_n)$, $\ldots$, $h_m(x_1, \ldots, x_n)$ are $(k-1)$-level compositions of basic functions.*

By induction over k, one can easily prove that all compositions of basic functions are homogeneous. For example:

- a linear combination is a basic function;
- an OWA combination of two values is a 2-level composition of basic functions;
- a general OWA operation is a 3-level composition of basic functions.

It turns out that an arbitrary homogeneous function can be approximated by appropriate 3-level compositions.

Definition 3. *Let $k > 0$ be a positive integer. We say that k-level compositions have a* universal approximation property for homogeneous functions *if for every continuous homogeneous function $f(x_1, \ldots, x_n)$, and for every two numbers $\varepsilon > 0$ and $\Delta > 0$, there exists a function $\widetilde{f}(x_1, \ldots, x_n)$ which is a k-level composition of basic functions and for which $|f(x_1, \ldots, x_n) - \widetilde{f}(x_1, \ldots, x_n)| \le \varepsilon$ for all $x_1, \ldots, x_n$ for which $|x_i| \le \Delta$ for all i.*

Theorem 1. *3-level compositions have a universal approximation property for homogeneous functions.*

(As we mentioned in Section 1, for readers' convenience, all the proofs are located in the special Proofs section.)

A natural question is: do we need that many levels of composition? What is we only use 1- or 2-level compositions? It turns out that in this case, we will not get the universal approximation property – and thus, the 3 levels of OWA operations is the smallest possible number.

Theorem 2

- *1-layer computations do not have a universal approximation property for homogeneous functions;*
- *2-layer computations do not have a universal approximation property for homogeneous functions.*

5 Why Linear Operations, min, and max: A Computational Justification

A natural question is: why should we select linear functions, min, and max as basic functions? One possible answer is that these operations are the fastest to compute, i.e., they require the smallest possible number of computational steps.

Indeed, the fastest computer operations are the ones which are hardware supported, i.e., the ones for which the hardware has been optimized. In modern computers, the hardware supported operations with numbers include elementary arithmetic operations ($+$, $-$, $\cdot$, $/$, etc.), and operations min and max.

In the standard (digital) computer (see, e.g., [2])

- addition of two n-bit numbers requires, in the worst case, $2n$ bit operations: n to add corresponding digits, and n to add carries;
- multiplication, in the worst case, means n additions – by each bit of the second factor; so, we need $O(n^2)$ bit operations;
- division is usually performed by trying several multiplications, so it takes even longer than multiplication;
- finally, min and max can be performed bit-wise and thus, require only n bit operations.

Thus, the fastest elementary operations are indeed addition (or, more generally, linear combination), min, and max.

6 Proof of Theorems 1 and 2

1. Before we start proving, let us notice that the values of the functions $\min(x_{i_1}, \ldots, x_{i_m})$ and $\max(x_{i_1}, \ldots, x_{i_m})$ depend on the order between the values $x_1, \ldots, x_n$. There are $n!$ possible orders, so we can divide the whole n-dimensional space of all possible tuples $(x_1, \ldots, x_n)$ into $n!$ zones corresponding to these different orders.
2. In each zone, a basic function is linear:

- a linear function is, of course, linear;
- a minimizing function $\min(x_{i_1}, \ldots, x_{i_m})$ is simply equal to the variable x_{i_k} which is the smallest in this zone and is, thus, linear;
- a maximizing function $\max(x_{i_1}, \ldots, x_{i_m})$ is simply equal to the variable x_{i_k} which is the largest in this zone and is, thus, also linear.

3. If a function $f(x_1, \ldots, x_n)$ can be approximated, with arbitrary accuracy, by functions from a certain class, this means that $f(x_1, \ldots, x_n)$ is a limit of functions from this class.
4. Basic functions are linear in each zone; thus, their limits are also linear in each zone. Since some homogeneous functions are non-linear, we can thus conclude that basic functions do not have a universal approximation property for homogeneous functions.
5. Let us now consider 2-level compositions of basic functions, i.e., functions of the type $f(x_1, \ldots, x_n) = g(h_1(x_1, \ldots, x_n), \ldots, h_m(x_1, \ldots, x_n))$, where g and h_i are basic functions.

Since there are three types of basic functions, we have three options:

- it is possible that $g(x_1, \ldots, x_m)$ is a linear function;
- it is possible that $g(x_1, \ldots, x_m)$ is a minimizing function; and
- it is possible that $g(x_1, \ldots, x_m)$ is a maximizing function.

Let us consider these three options one by one.

5.1. Let us start with the first option, when $g(x_1, \ldots, x_m)$ is a linear function. Since on each zone, each basic function h_i is also linear, the composition $f(x_1, \ldots, x_n)$ is linear on each zone.
5.2. If $g(x_1, \ldots, x_m)$ is a minimizing function, then on each zone, each h_i is linear and thus, the composition $f(x_1, \ldots, x_n)$ is a minimum of linear functions. It is known that minima of linear functions are concave; see, e.g., [8]. So, within this option, the function $f(x_1, \ldots, x_n)$ is concave.
5.3. If $g(x_1, \ldots, x_m)$ is a maximizing function, then on each zone, each h_i is linear and thus, the composition $f(x_1, \ldots, x_n)$ is a maximum of linear functions. It is known that maxima of linear functions are convex; see, e.g., [8]. So, within this option, the function $f(x_1, \ldots, x_n)$ is convex.
6. In each zone, 2-level compositions of basic functions are linear, concave, or convex. The class of all functions approximable by such 2-level compositions is the class of limits (closure) of the union of the corresponding three classes: of linear, concave, and convex sets. It is known that the closure of the finite union is the union of the corresponding closures. A limit of linear functions is always linear, a limit of concave functions is concave, and a limit of convex functions is convex. Thus, by using 2-level compositions, we can only approximate linear, concave, or convex functions. Since there exist homogeneous functions which are neither linear nor concave or convex, we can thus conclude that 2-level compositions are not universal approximators for homogeneous functions.
7. To complete the proof, we must show that 3-level compositions are universal approximators for homogeneous functions. There are two ways to prove it.
7.1. First, we can use the known facts about concave and convex functions [8]:

- that every continuous function on a bounded area can be represented as as a difference between two convex functions, and
- that every convex function can be represented as a maximum of linear functions – namely, all the linear functions which are smaller than this function.

These facts are true for general (not necessarily homogeneous) functions. For homogeneous functions $f(x_1, \ldots, x_n)$, one can easily modify the existing proofs and show:

- that every homogeneous continuous function on a bounded area can be represented as as a difference between two convex homogeneous functions, and
- that every homogeneous convex function can be represented as a maximum of homogeneous linear functions – namely, all the homogeneous linear functions which are smaller than this function.

Thus, we can represent the desired function $f(x_1, \ldots, x_n)$ as the difference between two convex homogeneous functions $f(x_1, \ldots, x_n) = f_1(x_1, \ldots, x_n) - f_2(x_1, \ldots, x_n)$. Each of these convex functions can be approximated by maxima of linear functions and thus, by 2-level compositions. Substraction $f_1 - f_2$ adds the third level, so $f(x_1, \ldots, x_n)$ can indeed be approximated by 3-level compositions.

To prove that a function $f(x_1, \ldots, x_n)$ can be represented as a different between two convex functions, we can, e.g., first approximate it by a homogeneous function which is smooth on a unit sphere $\{(x_1, \ldots, x_n) : x_1^2 + \ldots + x_n^2 = 1\}$, and then take $f_1(x_1, \ldots, x_n) = k \cdot \sqrt{x_1^2 + \ldots + x_n^2}$ for a large k. For smooth functions, convexity means that the Hessian matrix – consisting of its second derivatives $\dfrac{\partial^2 f}{\partial x_i \partial x_j}$ – is positive definite.

For sufficiently large k, the difference

$$f_2(x_1, \ldots, x_n) = f_1(x_1, \ldots, x_n) - f(x_1, \ldots, x_n)$$

is also convex – since its second derivatives matrix is dominated by positive definite terms coming from f_1. Thus, the difference $f_1 - f_2 = f$ is indeed the desired difference.

7.2. Another, more constructive proof, is, for some $\delta' > 0$, to select a finite δ'-dense set of points $e = (e_1, \ldots, e_n)$ on a unit square. For each such point, we build a 2-level composition which coincides with f on the corresponding ray $\{\lambda \cdot (e_1, \ldots, e_n) : \lambda > 0\}$. This function can be obtained, e.g., as a minimum of several linear functions which have the right value on this ray but change drastically immediately outside this ray.

For example, let $f_0(x)$ be an arbitrary homogeneous linear function which coincides with $f(x)$ at the point e – and thus, on the whole ray. To construct the corresponding linear functions, we can expand the vector e to an orthonormal basis e, e', e'', etc., and take linear functions $f_0(x) + k \cdot (e' \cdot x)$ and $f_0(x) - k \cdot (e' \cdot x)$ for all such e' (and for a large $k > 0$). Then, the minimum of all these functions is very small outside the ray.

We then take the maximum of all these minima – a 3-level composition.

The function $f(x_1, \ldots, x_n)$ is continuous on a unit sphere and thus, uniformly continuous on it, i.e., for every $\varepsilon > 0$, there is a δ such that δ-close value on the unit sphere lead to ε-close values of f. By selecting appropriate δ' and k (depending on δ), we can show that the resulting maximum is indeed ε-close to f.

The theorem is proven.

7 Conclusions

In multi-criteria decision making, it is necessary to aggregate (combine) utility values corresponding to several criteria (parameters). The simplest way to combine these values is to use linear aggregation. In many practical situations, however, linear aggregation does not fully adequately describe the actual decision making process, so non-linear aggregation is needed.

From the purely mathematical viewpoint, the next natural step after linear functions is the use of quadratic functions. However, in decision making, a different type of non-linearities are usually more adequate than quadratic ones: non-linearities like OWA or Choquet integral that use min and max in addition

to linear combinations. In this paper, we explain the empirically observed advantage of such aggregation operations. Specifically, we show that, due to the known properties of the utilities, reasonable operations for aggregating utilities must be scale-invariant. Aggregation operations like OWA or Choquet integral are scale-invariant, while quadratic functions are not. We also prove that operations like OWA and Choquet are general: to be more precise, we prove that compositions of linear functions, min, and max are universal approximators for scale-invariant operators.

Acknowledgments

This work was supported in part by NSF grant HRD-0734825, by Grant 1 T36 GM078000-01 from the National Institutes of Health, by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006-08, and by the Max Planck Institut für Mathematik.

The authors are very thankful to the anonymous referees for valuable suggestions.

References

1. Ceberio, M., Modave, F.: An interval-valued, 2-additive Choquet integral for multicruteria decision making. In: Proceedings of the 10th Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems IPMU 2004, Perugia, Italy (July 2004)
2. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge (2001)
3. Feynman, R., Leighton, R., Sands, M.: Feynman Lectures on Physics. Addison-Wesley, Reading (2005)
4. Grabisch, M., Murofushi, T., Sugeno, M. (eds.): Fuzzy Measures and Integrals. Physica-Verlag, Heidelberg (2000)
5. Keeney, R.L., Raiffa, H.: Decisions with Multiple Objectives. John Wiley and Sons, New York (1976)
6. Luce, R.D., Raiffa, H.: Games and Decisions: Introduction and Critical Survey. Dover, New York (1989)
7. Raiffa, H.: Decision Analysis. Addison-Wesley, Reading (1970)
8. Rockafeller, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
9. Yager, R.R., Kacprzyk, J. (eds.): The Ordered Weighted Averaging Operators: Theory and Applications. Kluwer, Norwell (1997)

Building an Affective Model for Intelligent Tutoring Systems with Base on Teachers' Expertise

Yasmín Hernández[1], L. Enrique Sucar[2], and Gustavo Arroyo-Figueroa[1]

[1] Instituto de Investigaciones Eléctricas, Gerencia de Sistemas Informáticos
Reforma 113, Col. Palmira, 62490, Cuernavaca, Morelos, México
{myhp,garroyo}@iie.org.mx
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica, Coord. Ciencias Computacionales
Luis Enrique Erro # 1, Tonantzintla, Puebla, México
esucar@inaoep.mx

Abstract. We are developing an affective behavior model for intelligent tutoring systems. This model provides students with a suitable response considering the knowledge and affective state of the student. The affective behavior model comprises two components: an affective student model and an affective tutor model. The affective student model is based on the OCC cognitive model of emotions. The main contribution of this work is the affective tutor model, which is built with base on the expertise of a group of teachers. For this end, we conducted a study to ask teachers how they deal with affective aspects when they are teaching; and we used the results of the study to refine our model. The affective behavior model relies on dynamic Bayesian networks and dynamic decision networks. In this paper, we present the affective behavior model built based on the teachers´ responses, its application to a robotics tutor and an evaluation with a Wizard of Oz study.

Keywords: Affective student model, affective tutor, intelligent tutors.

1 Introduction

In last years it has been recognized the importance of emotions in motivation and learning; and it has been assumed that if emotions are considered in intelligent tutoring systems we can obtain better outcomes in learning [1]. Towards this goal, we are developing an affective behavior model for intelligent tutoring systems. The objective of the model is to provide students with a suitable action according to her affective and knowledge states. The affective behavior model is composed by an affective student model and an affective tutor model. The affective student model has been designed with base on personality and emotion theories, and relies on a probabilistic network, based on the approach proposed in [2]. The affective tutor model is being built with base on teachers' expertise. Following [2], the tutorial actions are selected according to the pedagogical and affective state of the student by using a dynamic decision network with a utility measure on both, learning and affect. To build the affective tutor model we conducted a study with nine experienced teachers; the results from the study helped us to validate our assumptions and refine the model. The aim of

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 754–764, 2008.

this study was to know which actions the teachers do according to the affective and pedagogical student state, and why they select the actions. This is one of the first attempts to build an affective tutor, in particular based on an extensive study with teachers. There are very few studies reported with so many teachers [3] in the literature. Moreover, we have conducted an initial evaluation by means of a Wizard of Oz study; we applied the affective behavior model in a tutor for robotics. In this study, 20 students participated; the results show a good agreement between the student model predictions and the affective state, and in general the students' opinions on the affective actions are positive.

In this paper we describe the affective behavior model, present the study conducted to build the affective tutor model and present the Wizard of Oz study. The rest of the paper is organized as follows: section 2 describes the affective behavior model; next, section 3 describes the test domain; in section 4 we present the teachers study; section 5 presents the Wizard of Oz study and its results; finally, we outline our outgoing and future work, and present the conclusions.

2 Affective Behavior Model

The importance of the affective state in motivation and learning has been recognized, and it has been assumed that if the affective state is considered in *intelligent tutoring systems* (ITS), we can get better outcomes in learning. Towards this goal, we are developing an *affective behavior model* (ABM) for intelligent tutoring systems, to provide students with a tutorial action which considers the affective and knowledge state of students. For affective behavior we mean two aspects: i) to infer the affective student state; and ii) to establish the tutorial action considering the student's affect. A diagram of the ABM is presented in fig.1.

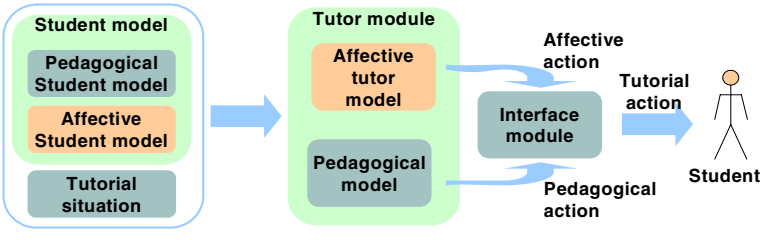


Fig. 1. General block diagram of the affective behavior model. The model is composed by an affective student model and an affective tutor model. The tutor model produces an affective action taking into account the student model and the tutorial situation. The affective action is a component of the tutorial action to be presented to the student.

The ABM has two components: the *affective student model* and the *affective tutor model*. The affective student model is part of the student model and the affective tutor model is part of the tutor module. To establish the next tutorial action to be presented to students, the ABM considers the affective student model, the pedagogical student model and the tutorial situation. The affective action gives affective information to the pedagogical tutor model to establish the next pedagogical action, and also gives

affective information to the interface module to establish the physical realization of the pedagogical action. Hence, the student receives a tutorial action composed by an affective and a pedagogical action. The strategy of first selecting the affective action, and then to use it to guide the selection of the pedagogical actions, is based on feedback from the teachers in our investigations. Some teachers said that they first consider the affective state, and then they evaluate the knowledge state. Thus, the affective behavior model will determine the tutorial action to be delivered to the student. A tutorial action is composed by a pedagogical action and an affective action.

2.1 Affective Student Model

The first component of the ABM is the affective student model. Our affective student model uses the OCC Cognitive Model of Emotions [4] to assess students' emotions. The OCC model defines emotion as the outcome of a cognitive appraisal of the current situation with respect to one's goals. The affective student model is represented as a Bayesian network which is shown in Fig. 2, in a high-level view.

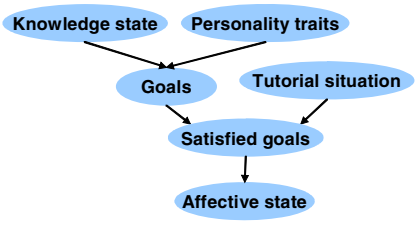


Fig. 2. High level Bayesian network for the affective student model. Each node in this network represents a group of nodes in the detailed model.

The affective state is not static; but it changes over time as a result of the changing environment and the particular interpretation of the situation for each individual. The student's appraisal of the current situation given her goal is represented by the relation between the *goals* and *tutorial situation* nodes through the *satisfied goals* node. The influence of the appraisal process on the student's affect is represented by the link between the *satisfied goals* node and the *affective state* node. From the complete set of emotions proposed by the OCC model, the affective model only includes six emotions: *joy*, *distress*, *pride*, *shame, admiration* and *reproach*. They are represented as three pairs of mutually exclusive emotions: *joy-distress*, *pride-shame* and *admiration-reproach,* each represented by a binary node in the detailed network.

According to the OCC model, the goals are fundamental to determine the affective state. We established three goals for our domain 1) *to learn the topics*, 2) *to have success*, and 3) *to be as fast as possible.* Each goal is represented by a binary node in the detailed network. In our model, goals are inferred from indirect sources of evidence. We use personality traits and student's knowledge as a predictor of a student's goals. Personality traits are based on the Five Factor Model [5,6], which considers five dimensions for personality: *openness*, *conscientiousness*, *extraversion*, *agreeableness*, and *neuroticism*. Currently, we use only two of them (*conscientiousness* and *neuroticism*) to establish goals because a relationship has been established between these two dimensions and learning [7]. Although we use only two personality dimensions to infer

goals; we think that all five dimensions influence the affective state. The information for the knowledge state and tutorial situation nodes comes from the pedagogical student model and the student action results. The dependency relations in the Bayesian dynamic network have been established based on the literature [6], questionnaires applied to students and intuition.

2.2 Affective Tutor Model

Once the affective student state has been obtained, the tutor has to respond accordingly. The tutor needs an affective model which establishes parameters that enable a mapping from the affective and pedagogical student state to tutorial actions. In this way, the second component of the ABM is the affective tutor model. The tutorial actions are composed by a pedagogical action and an affective action. The affective action tries to promote a positive affective student state, and the pedagogical action tries to convey knowledge that the student needs to know.

We consider as affective actions the way in which the pedagogical content is delivered to the student; e.g., the words, the facial expression, colors or sound included in the message. In the work presented here, an affective action is an animation of a pedagogical agent who delivers the tutorial actions to students. Although there are other ways to deliver affective actions, in this study we restricted them to the character animations; so we can clearly separate the affective actions from the pedagogical ones, and to facilitate the evaluation of both components. Our main hypothesis is that the tutor action has a direct influence on learning and on the affective state of the student; and by selecting the appropriate tutorial action (i.e. according to the current student state), the tutor could improve the learning process and the affective state of the students. Given this hypothesis, we want to help students to learn and at the same time to foster a positive affective state. Towards this aim, we use decision theory considering a trade-off between learning and affect. Fig. 3 depicts a representation of the affective tutor model as a dynamic decision network (DDN).

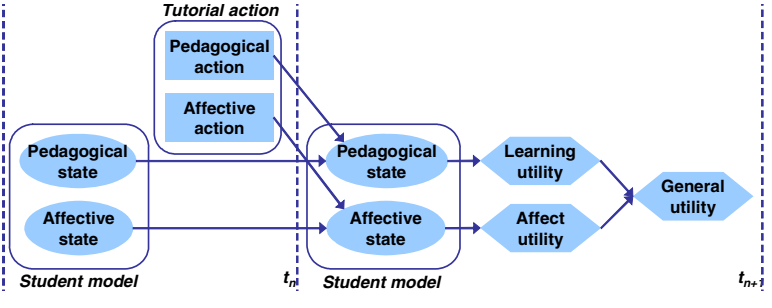


Fig. 3. High level decision network for the affective tutor model. This network is used to establish the tutorial action to be presented to the student.

The dynamic Bayesian network implicit in the DDN, is used to predict how the tutorial actions influence the affective and pedagogical state of the student, considering the current affective and pedagogical states. This prediction is used to establish the utility of each tutorial action for the current state.

We consider the effects of the tutorial actions in each state individually, and then we consider a general effect in the tutoring process. The DDN establishes the tutorial action considering two utility measures, one on learning and other on affect, which are combined to obtain the global utility by a weighted linear combination.

At the time t_n we have the current student state and the possible tutorial actions. When the student makes an action, i.e. after the student model is updated, a new time slice is added; thus at time t_{n+1} we have the prediction of how the tutor action influences the student affective and pedagogical state, from which we estimate the individual and global utilities. The *affective state* and *pedagogical state* nodes in the high level DDN at time t_n (Fig. 3) are actually Bayesian networks.

The Bayesian network for the affective state was described in last section (for more detailed description see [8]) and the Bayesian network for the pedagogical state is described in [9]. The influence of each tutorial action on the pedagogical and affective state, and the utility of the student's states are established according to the tutor preferences, which are based on the expertise of a group of teachers, according to the teachers study described in section 4. For learning, we measure the utility in terms of how much learning could be increased with the tutorial action given the current pedagogical state. The utility on learning is always a number greater or equal than 0, because currently we do not model forgetting. In the same way, for the affective state, we measure the utility in terms of how much the affect could be improved with the tutorial action given the current affective state. In this case, the utility can be a number lower than 0 because the tutorial action can have a negative effect on the affective state of the student. Finally, for the overall utility, we obtain a weighted sum of the utility on learning and the utility on affect. In this way, the tutor calculates the utility for each tutorial action considering the current state and it selects the tutorial action with the maximum expected utility. When the tutorial action is selected, the DDN has finished its work and the time slice t_{n+1} is discharged. This is because, currently, the tutorial action is not used to update the student model but only to predict the impact of the tutorial action. At this point, the tutor delivers the selected action to the student; waits for the next student action and the student model is updated another time.

Although the general structure of the ABM has been defined, establishing the parameters of the model and, in particular, the appropriate tutorial actions, are a complex problem. For this we conducted a study with a group of teachers to establish the affective actions and parameterize the model. Before we present the study, we briefly describe the domain in which the model is applied.

3 Learning Environment for Robotics

We are in the process of integrating the ABM into an ITS for learning mobile robotics [9]. In this environment, the students interacting with the ITS perform experiments with a mobile robot using a virtual laboratory. The tutor gives instruction to the students based on a probabilistic representation of the students' pedagogical state. The pedagogical actions are explanations about the topics related to the current experiment in the virtual laboratory. This ITS does not have an animated agent to present the

pedagogical action to the students; therefore, the integration of the ABM into this ITS includes to add an animated pedagogical agent to present the tutorial actions in a more personalized and natural way. In fig. 4, a screenshot of the ITS and an example of a pedagogical action are presented.

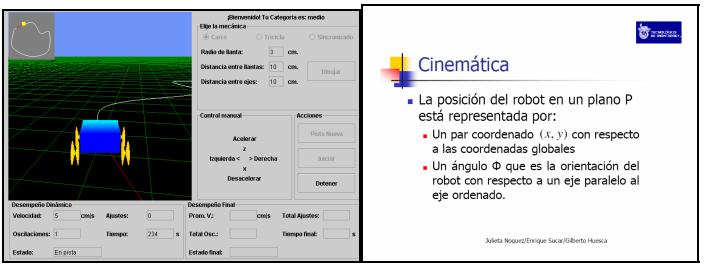


Fig. 4. Left: Screenshot of the ITS for robotics, a student is performing an experiment on line following. Right: Example of a lesson presented to the student as a pedagogical action.

4 Teachers' Study

We conducted a study with nine professors to validate our assumptions and refine our affective model. The participating professors have taught in several scholar levels, from high school to PhD, for 12.4 years in average. The aim of the study was i) to know which actions the teachers do according to the affective and pedagogical student state; ii) to know why they select that actions, iii) to select an animated character to be integrated into the ITS; and iv) to select the appropriate animations to be used as affective actions. The study had two parts. In the first part the teachers were presented with four characters of Microsoft Agent and every possible animation displayed for the characters of Microsoft Agent, so they can see the potential of the animated characters. In Fig. 5 the characters of Microsoft Agent used in the study are shown.

Fig. 5. Characters of Microsoft Agent. Left to right: Robbie, Genie, Peedy and Merlin.

A program to show the agents and its animations was developed; the teachers could select each of the characters and see all of its animations as many times as they wanted. In this phase of the survey, the teachers selected a character to be included in the ITS to present the tutorial actions. Also, they selected the animations they thought are suitable to be shown to students as affective actions. The animations selected in this phase were used in the next phase of the study.

Fig. 6 shows a screenshot of the program used in this phase of the study. The participants could choose the character in the file menu; the selected character, Robbie, is shown next to the window, right side in Fig. 6. A list of the available animations is presented; in this way the teachers could select one of them and the character plays the selected animation.

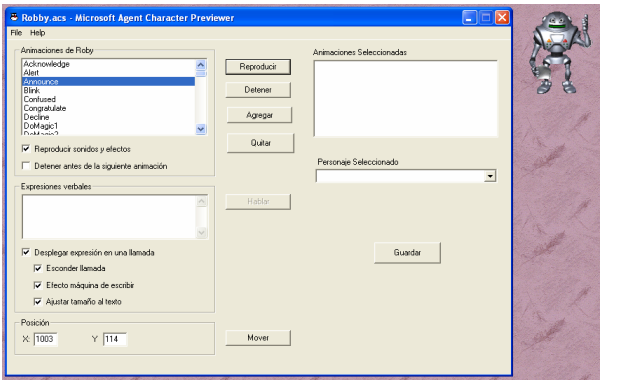


Fig. 6. Screenshot of the program to see the characters and animations of Microsoft Agent. In the file menu the teachers can selected the character. In the combo list on the left they can select the animations to be play by the character, and when they found an interesting animation they can select it to add it to the combo list on the right.

As result of this phase of the study, we obtained that seven teachers selected the character Robbie of Microsoft Agent, and two teachers selected the character Merlin. The teachers that selected Robbie said that Merlin is a more expressive character but they agreed that Robbie is more suitable for the subject matter, and for the age of the students.

In the second part, the teachers interacted with the ITS for robotics. They could observe the ITS (without the ABM) and the possible pedagogical actions of the ITS. Then a video of students interacting with the ITS was presented to the teachers; they watched six different experiments carried out by two students. The participants were provided with facilities to replay, forward and backward the video.

The teachers were asked to establish the pedagogical action and the affective action they consider adequate for each situation and for the student affective and pedagogical state. We also asked teachers why they think the pedagogical and affective actions improve the affective and pedagogical states. The teachers' annotations were required for each experiment. In fig. 7 we present a screenshot of the program used in this second phase of the study.

The teachers were asked to rate the affective state and the pedagogical state of the students after each experiment. They also were asked to establish which animation they will use as an affective action and to establish the pedagogical action to present to the student in that context. Moreover, they were asked to say how they think the affective and the pedagogical actions selected would help the student to have a better affective

state and to improve her learning process. In a similar way as the computerized tutor, the teachers have to infer the students' pedagogical and affective states from the results of the students' interaction in the experiments. We think that in this way the results of the study are more appropriate to be applied for building the affective tutor, as the conditions are analogous.

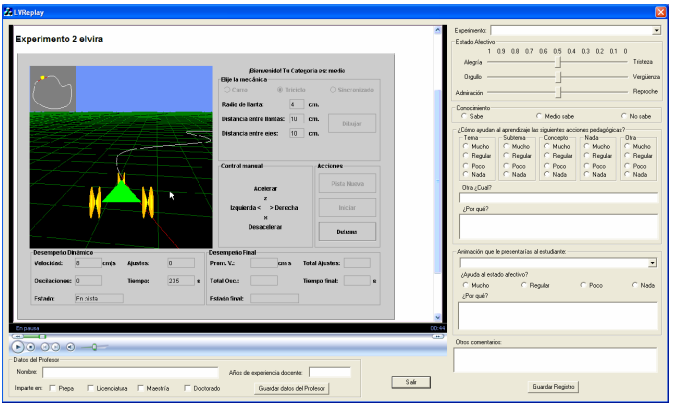


Fig. 7. Interface for the second phase of the teachers' study. Left: Video of an experiment being performed in the learning environment for robotics. Right and bottom: options for the teachers. The teachers were asked to establish the affective and pedagogical action to be presented to students taking into account the affective and knowledge state, according to the experiment in the video.

In this phase, we obtained eight animations the teachers selected according to the affective state of the students; these are shown in table 1; also, the average rate for each emotion when the teachers selected the action is presented. The emotions are rated from **1** to **100**, where **100** means *joy*, *pride* or *admiration* and **1** means *distress*, *shame* or *reproach*. This information is used to establish the conditional probabilities for the DDN for the affective tutor model (see section 2). In Fig. 8 we depict the Robbie's animations selected by the teachers.

Table 1. Animations preferred by the teachers. Teachers selected the animation on the left when the affect were in the levels on the right columns. Only the positive emotions are shown, the negative emotions are their complements.

Affective Action	Joy	Pride	Admiration
Acknowledge	62.25	54.00	52.25
Announce	60.00	73.00	50.00
Congratulate	72.69	72.31	70.00
Confused	26.43	29.57	28.57
GetAttention	54.33	55.67	54.67
Explain	77.00	70.00	50.00
Suggest	55.29	52.86	56.86
Think	42.67	35.33	46.33

We also obtained why the teachers consider the different pedagogical actions (lessons) and affective actions (animations) adequate for the pedagogical and affective student state. For example, when the student fails in an experiment, the student seems not to know the topics in the experiment, therefore the pedagogical action selected by the teachers is an overall lesson about the topics of the experiment, and the affective action could be the animation *explain*. The animations *acknowledge*, *announce* and *congratulate* are used when the tutorial situation is positive; the animations *confused* and *getAttention* are used when the tutorial situation is regular; and the animations *explain*, *suggest* and *think* are used when the tutorial situation is negative.

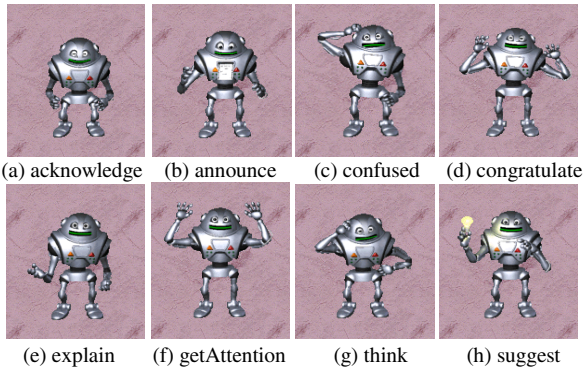


Fig. 8. Robbie's animations selected by the teachers. Between more than 50 animations available for Robbie the teachers selected eight animations to be used as affective actions.

The results of this study were used to refine the model and to set the parameters for the affective behavior model to be applied to the ITS for robotics. In the next section, the evaluation of the model by a Wizard of Oz study with a group of students is presented.

5 Wizard of Oz Study

After we completed the ABM based on the results of the teachers' survey, we wanted to evaluate the performance of the affective model previous to the completion of the integration of the ABM into the ITS. Therefore, we conducted a Wizard of Oz study. This kind of study has proved to be useful for evaluation and design of systems which are in the first stages of development. In this type of studies, a person (or a mechanism) simulates the system components which are not finished [10]. We had 20 students participating in the study; one undergraduate student and 19 graduate students.

The students answered a personality test, based on the five factor model [11]. The test is composed by 50 adjectives, and the students have to rate how each adjective applies to them. With this information we can establish the personality traits of the students and their goals in the ITS.

The study consisted in presenting to the students three different tutorial scenarios and the affective and pedagogical action selected by the affective behavior model

considering the affective student state and the tutorial situation presented in the scenario. Robbie, the animated character, presented the actions.

Students have to rate their affective state given the tutorial situation before and after they received the tutorial actions. Also they had to state if the tutorial actions helped them to learn, and state why they think so.

We compared the affective state reported by the students with the affective state established by the affective student model. We found that for the emotion *joy-distress*, the model established the affective state correctly 72% of the cases; for the emotion *pride-shame*, the model established the affective state correctly 70% of the cases; and for the emotion *admiration-reproach*, the model established the affective state correctly 55% of the cases.

Moreover, we found that 65% of students considered that their general affective state improved after the tutorial action and 90% of the student considered that their pedagogical state improved after the tutorial action, these results are summarized in table 2.

Table 2. Students' opinions on how their affective and pedagogical state changed after received the tutorial actions established by the affective behavior model

State	Affective state		Pedagogical state	
	Students	Percentage	Students	Percentage
Better	13	65%	18	90%
Same	2	10%	1	5%
Worst	5	25%	1	5%

These results are promising, as they show in general a good agreement in terms of the affective state of the students, and a positive opinion of the affective and pedagogical actions. However, a controlled user evaluation comparing the learning differences with and without the affective component is required to confirm the hypothesis that this has a positive impact on learning; this is the next phase in our research.

6 Summary and Future Work

We have developed an affective behavior model for ITS. The affective behavior model integrates an affective student model based on the OCC cognitive model of emotions and relies on a DBN. The tutorial actions are selected according to the pedagogical and affective state by using a DDN with a utility measure on both, learning and affect. The model was developed based on the results from a teachers' study which helped to establish the pedagogical and affective actions, as well as the model parameters.

The ABM was evaluated in the robotics domain via a Wizard of Oz experiment. The results are encouraging, as they show a relatively high agreement between the affective states given by the model with those of the students. Also, in general, the perception of the students is that the tutor improves their pedagogical and affective state. The next step is to complete the integration of the ABM with the ITS for learning mobile robotics; and then to conduct another user study.

We have also applied the ABM in an educational game for number factorization for elementary school students. In this domain we used the Merlin character [12], and

we have conducted a controlled evaluation with a group of students [13]. We evaluated the learning gains in control and experimental groups, and we found that the students who played with the affective game learnt more than the other group. These results, together with the current study in a different domain, give strong evidence in favor of the proposed affective tutor model.

Our main contribution is the development of a decision-theoretic affective tutor model, which selects a tutorial action based on the affective and the pedagogical state of the student; and the refinement of this model based on the opinions of experienced teachers.

References

1. Qu, L., Wang, N., Johnson, W.L.: Using Learner Focus of Attention to Detect Learner Motivation Factors. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 70–73. Springer, Heidelberg (2005)
2. Conati, C., Mclaren, H.: Data-driven Refinement of a Probabilistic Model of User Affect. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 40–49. Springer, Heidelberg (2005)
3. Alexander, S., Hill, S., Sarrafzadeh, A.: How do Human Tutors Adapt to Affective State? In: 10th International Conference on User Modeling, UM 2005, Workshop on Adapting the Interaction Style to Affective Factors, Edinburgh, Scotland, UK, July 24-29 (2005)
4. Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press, Cambridge (1988)
5. Costa, P.T., McCrae, R.R.: Four Ways Five Factors are Basic. Personality and Individual Differences 13(1), 653–665 (1992)
6. Boeree, G.: Personality Theories, Electronic Textbook (1998) (Retrieved on February 2005), http://www.ship.edu/~cgboeree/
7. Heinström, J.: The impact of personality and approaches to learning on information behaviour. Information Research 5(3) (April 2000), http://informationr.net/ir
8. Hernández, Y., Arroyo-Figueroa, G., Sucar, L.E.: Intelligent Tutoring System with Affective Behavior. In: Proceedings of 6th Mexican International Conference on Artificial Intelligence, MICAI 2007, Aguascalientes, México, November 4-10, 2007. IEEE CS Press, Los Alamitos (2007)
9. Noguez, J., Sucar, E.: A Semi-open Learning Environment for Virtual laboratories. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 1185–1194. Springer, Heidelberg (2005)
10. Dow, S., MacIntyre, B.: New Media Collaboration through Wizard-of Oz Simulations. In: Computer-Human Interaction 2007, Workshop on HCI and New Media Arts: Methodology and Evaluation, San José, California, Estados Unidos, April 28-May 3 (2007)
11. Boeree, G.: Big Five Minitest, On-line test (Retrieved on February 2005), http://www.ship.edu/~cgboeree/
12. Hernández, Y., Sucar, E.: User study to evaluate an affective behavior model in an educational game. In: 13th International Conference on Artificial Intelligence in Education, Workshop in Modeling and Scaffolding Experiences to Impact Learning, California, USA, July 9-13 (2007)
13. Hernández, Y., Sucar, E., Conati, C.: An Affective Behavior Model for Intelligent Tutors. In: Woolf, B.P., et al. (eds.) ITS 2008. LNCS, vol. 5091, pp. 819–821. Springer, Heidelberg (2008)

A Knowledge Discovery Framework for Learning Task Models from User Interactions in Intelligent Tutoring Systems

Philippe Fournier-Viger[1], Roger Nkambou[1], and Engelbert Mephu Nguifo[2]

[1] University of Quebec in Montreal, Montreal (QC), Canada
[2] Université Lille-Nord de France, Artois, F-62307 Lens, CRIL, F-62307 Lens
CNRS UMR 8188, F-62307 Lens, France
fournier_viger.philippe@courrier.uqam.ca, nkambou.roger@uqam.ca,
mephu@cril.univ-artois.fr

Abstract. Domain experts should provide relevant domain knowledge to an Intelligent Tutoring System (ITS) so that it can guide a learner during problem-solving learning activities. However, for many ill-defined domains, the domain knowledge is hard to define explicitly. In previous works, we showed how sequential pattern mining can be used to extract a partial problem space from logged user interactions, and how it can support tutoring services during problem-solving exercises. This article describes an extension of this approach to extract a problem space that is richer and more adapted for supporting tutoring services. We combined sequential pattern mining with (1) dimensional pattern mining (2) time intervals, (3) the automatic clustering of valued actions and (4) closed sequences mining. Some tutoring services have been implemented and an experiment has been conducted in a tutoring system

1 Introduction

Domain experts should provide relevant domain knowledge to an Intelligent Tutoring System (ITS) so that it can guide a learner during problem-solving activities. One common way of acquiring such knowledge is to use the method of cognitive task analysis that aims at producing effective problem spaces or task models by observing expert and novice users for capturing different ways of solving problems. However, cognitive task analysis is a very time-consuming process [1] and it is not always possible to define a satisfying complete or partial task model, in particular when a problem is ill-structured. According to Simon [2], an ill-structured problem is one that is complex, with indefinite starting points, multiple and arguable solutions, or unclear strategies for finding solutions. Domains that include such problems and in which, tutoring targets the development of problem-solving skills are said to be ill-defined (within the meaning of Ashley et al. [3]). An alternative to cognitive task analysis is constraint-based modeling (CBM) [4], which consist of specifying sets of constraints on what is a correct behavior, instead of providing a complete task description. Though this approach was shown to be effective for some ill-defined domains, a domain expert has to design and select the constraints carefully.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 765–778, 2008.

Contrarily to these approaches where domain experts have to provide the domain knowledge, a promising approach is to use knowledge discovery techniques for automatically learning a partial problem space from logged user interactions in an ITS, and to use this knowledge base to offer tutoring services.

We did a first work in this direction [5] by proposing a framework to learn a knowledge base from user interactions in procedural and ill-defined domains [5]. The framework takes as input sequences of user actions performed by expert, intermediate and novice users, and consist of applying two knowledge discovery techniques. First, sequential pattern mining (SPM) is applied to discover frequent action sequences. Then, association rules discovery find associations between these significant action sequences, relating them together. The framework was applied in a tutoring system to extract a partial problem space that is used to guide users, and thus showed to be a viable alternative to the specification of a problem-space by hand for the same domain [5, 7]. This framework differs from other works that attempt to construct a task model from logged student interactions such as [8], [9] and [10], since these latter are devoid of learning, reducing these approaches to simple ways of storing or integrating raw user solutions into structures.

Although the framework [5] was shown to be useful, it can be improved in different ways. Particularly, in this paper, we present an extended SPM algorithm for extracting a problem space that is richer and more adapted for supporting tutoring services. This work was done following our application of the framework in the RomanTutor tutoring system [5].

The rest of the paper is organized as follow. First, it introduces RomanTutor [6] and the problem of SPM from user actions. Then it presents the limitations of the framework encountered and extensions to address these issues. Finally, it presents preliminary results of the application of the improved framework in RomanTutor, future work and a conclusion.

2 The RomanTutor Tutoring System

RomanTutor [6] (cf. fig. 1) is a simulation-based tutoring system to teach astronauts how to operate Canadarm2, a 7 degrees of freedom robot manipulator deployed on the International Space Station (ISS). During the robot manipulation, operators do not have a direct view of the scene of operation on the ISS and must rely on cameras mounted on the manipulator and at strategic places in the environment where it operates. The main learning activity in RomanTutor is to move the arm to a goal configuration. To perform this task, an operator must select at every moment the best cameras for viewing the scene of operation among several cameras mounted on the manipulator and on the space station.

In previous work [7], we attempted to model the Canadarm2 manipulation task with a rule-based knowledge representation model. Although, we described high-level rules such as to set the parameters of cameras in a given order, it was not possible to go in finer details to model how to rotate the arm joint(s) to attain a goal configuration. The reason is that for a given robot manipulation problem, there are many possibilities for moving the robot to a goal configuration and thus, it is not possible to define a complete and explicit task model. In fact there is no simple 'legal move generator' for

finding all the possibilities at each step. Hence, RomanTutor operates in an ill-defined-domain. As a solution, we identified 155 actions that a learner can take, which are (1) selecting a camera, (2) performing a small/medium/big increase or decrease of the pan/tilt/zoom of a camera and (3) applying a small/medium/big positive/negative rotation value to an arm joint. Then, we applied SPM to mine frequent action sequences from logged users' interactions [5]. The resulting knowledge base served in RomanTutor to track the patterns that a learner follows, and to suggest the next most probable actions that one should execute.

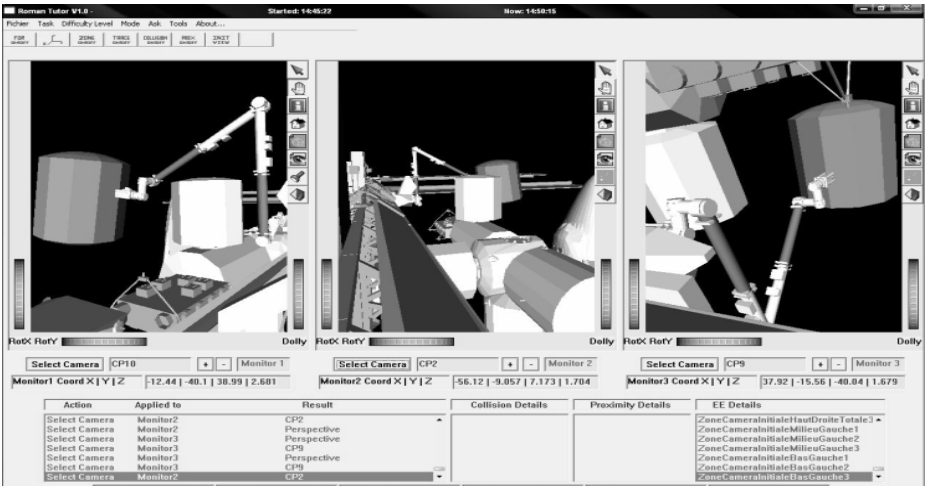


Fig. 1. The RomanTutor User Interface

3 Sequential Patterns Mining from User Actions

The problem of mining sequential patterns is stated as follows [11]. Let D be a transactional database containing a set of transactions (here also called plans) and a set of sequence of items (here called actions). An example of D is depicted in figure 2.a. Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set of actions. We call a subset X A an actionset and |X|, its size. Each action in an actionset (enclosed by curly brackets) are considered simultaneous. A sequence $s = (X_1, X_2, \ldots, X_m)$ is an ordered list of actionsets, where X_i A, $i \in \{1,\ldots,m\}$, and where m is the size of s (also noted |s|). A sequence $s_a = (A_1, A_2, \ldots, A_n)$ is contained in another sequence $s_b = (B_1, B_2, \ldots, B_m)$ if there exists integers $1 \leq i_1 < i_2 < \ldots < i_n \leq m$ such that $A_1 \quad B_{i1}$, $A_2 \quad B_{i2}$, . . . , $A_n \quad B_{in}$. The relative support of a sequence s_a is defined as the percentage of sequences $s \in D$ that contains s_a, and is denoted by $supD(s_a)$. The problem of mining sequential patterns is to find all the sequences s_a such that $supD(s_a) \geq$ *minsup* for a database D, given a support threshold *minsup*.

Consider the dataset of figure 2.a. The size of the plan 2 is 6. Suppose we want to find the support of S2. From figure 2.a, we know that S2 is contained in plan 1, 2 and 5. Hence, its support is 3 (out of a possible 6), or 0.50. If the user-defined minimum

support value is less than 0.50, then S2 is deemed frequent. To mine sequential patterns several algorithms have been proposed [11, 12, 13]. In our first experiment in RomanTutor, we chose PrefixSpan [12] as it is a promising approach for mining large sequence databases having numerous patterns and/or long patterns, and also because it can be extended to mine sequential patterns with user-specified constraints. Figure 2.b shows some sequential patterns extracted by PrefixSpan from the data in figure 2.a using a minimum support of 25%. In RomanTutor, one mined pattern is for example, to select the camera 6, which gives a close view of the arm in its initial position, slightly decrease the yaw of camera 6, select the elbow joint and decrease a little bit its rotation value. Although, the set of patterns extracted for RomanTutor constitutes a useful problem space that capture different ways of solving problems, we present next the limitations of SPM encountered in our first experiment, and extensions to PrefixSpan to address these issues.

ID	Sequences of actions
1	1 2 25 46 48 {9 10 11 31}
2	1 25 46 54 {10 11 25} 48
3	1 2 3 {9 10 11 31} 48
4	2 3 25 46 11 {14 15 48} 74
5	4 1 25 27 46 48
6	1 3 44 45 46 48

→

ID	Seq. patterns	Support
S1	1 46 48	66 %
S2	1 25 46 48	50 %
S3	1 25 46 {10 11}	33 %
S4	1 {9 10 31}	33 %
S5	1 {9 11 31}	33 %
…	…	…

Fig. 2. (a) A Data Set of 6 Plans (b) Example of Sequential Patterns Extracted

4 Extending Sequential Pattern Mining with Time Intervals

A first limitation that we encountered is that extracted patterns often contain "gaps" with respect to their containing sequences. For instance, in the example of figure 2, action "2" of plan 1 has not been kept in S1. A gap of a few actions is ok in a tutoring context because it eliminates non-frequent learners' actions. But when a sequence contain many or large gap(s), it becomes difficult to use this sequence to track a learner's actions and to suggest a next relevant step. Thus, there should be a way of limiting the size of the gaps in mined sequences. Another concern is that some patterns are too short to be useful in a tutoring context (for example, sequences of size 1). In fact, there should be a way of specifying a minimum sequential pattern size.

An extension of SPM that overcomes these limitations is to mine patterns from a database with time information. A time-extended database is defined as a set of time-extended sequences $s = <(t_1,X_1), (t_2,X_2),\ldots, (t_n,X_n)>$, where each actionset X_x is annotated with a timestamp tx. Each timestamp represents the time elapsed since the first actionset of the sequence. Actions within a same actionset are considered simultaneous. For example, one time-extended sequence could be <(0, a), (1, b c), (2, d)>, where action d was done one time unit after b and c, and two time units after a. The time interval between two actionsets (t_x,X_x) and (t_y,X_y) is calculated as $|t_x - t_y|$. In this work, we suppose a time interval of one time unit between any adjacent actionsets, so that $|t_x - t_y|$ become a measure of the number of actionsets between (t_x, X_x) and (t_y, X_y). The problem of Generalized Sequential Pattern Mining with Time Intervals (GSPM) [14] is to extract all time-extended sequences s from a time-extended database, such that

supD(s) ≥ minsup and that s respect all time constraints. Four types of constraints are proposed by Hirate and Yamana [14]. The constraints C1 and C2 are the minimum and maximum time interval required between two adjacent actionsets of a sequence (gap size). The constraints C3 and C4 are the minimum and maximum time interval required between the head and tail of a sequence. For example, for the sequence <(0, a), (1, b c), (2, d)>, the time interval between the head and the tail is 2 and the time interval between the actionset (0, a) and (1, b c) is 1.

Hirate and Yamana [14] have proposed an extension of the PrefixSpan algorithm for the problem of GPSM. We present it below –with slight modifications- as it is the basis of our work. The algorithm finds all frequent time-extended sequences in a database ISDB that respect minsup, C1, C2, C3 and C4, by performing a depth-first search. The algorithm is based on the property that if a sequence is not frequent, any sequence containing that sequence will not be frequent. The algorithm proceeds by recursively projecting a database into a set of smaller projected databases. This process allows growing patterns one action at a time by finding locally frequents actions.

In the following, the notation ISDB|(t,i) represents the time-extended database resulting from the operation of projecting a time-extended database ISDB with a pair (timestamp, item). ISDB|(t, i) is calculated as follow.

```
ISDB((t,i))
  ISDB|(t,i) := Ø.
  FOR each sequence σ=<(t1,X1),(t2,X2)…(tn,Xn)> of
ISDB.
  FOR each actionset (tx,Xx) of σ containing i.
    IF Xx/{i} = Ø
      s :=<(t_{x+1}-t_x,a_{x+1}), … (t_n-t_x,X_n)>
    ELSE
      s :=<(0, X_x/{i}), (t_{x+1}-t_x, a_{x+1}), … (t_n-t_x, X_n)>
    IF s ≠ Ø  and s satisfies C1, C2 and C4
      Add s to ISDB|(t,i).
  Return ISDB|(t,i).
```

The Hirate-Yamana algorithm (described below) discovers all frequent time-extended sequences.

```
algoHirate(ISDB, minsup, C1, C2, C3, C4)

  R := ø.

  Scan ISDB and find all frequent items with support
   higher than minsup.
  FOR each frequent item i,
    Add (0, i) to R.
    algoProjection(ISDB|(0, i), R, minsup,C1,C2,C3,C4).

  RETURN R;

  algoProjection(ISDB|prefix, R, minsup, C1, C2, C3, C4)
    Scan ISDB|prefix to find all pairs of item and
     timestamp, denoted (t, i) satisfying minsup, C1 and
  C2.
```

```
    FOR each pair (t, i) found
      newPrefix := Concatenate(prefix, (t, i)).
      IF newPrefix satisfies C3 and C4
        Add newPrefix to R.
        IF (size of ISDB|newPrefix) >= minsup
         algoProjection(ISDB|newPrefix, R, minsup,C1,C2
C3,C4).
```

To illustrate the Hirate-Yamana algorithm, let's consider applying it to the database ISDB depicted in figure 3, with a *minsup* of 50 % and the constraint C2 equals to 2 time units. In the first part of *algoHirate*, frequent actions *a*, *b* and *c* are found. As a result <(0,a)>, <(0,b)> and <(0,c)> are added to *R*, the set of sequences found. Then, for *a*, *b* and *c*, the projected database ISDB|(0,a), ISDB|(0,b) and ISDB|(0,c) are created, respectively. For each of these databases, the algorithm *algoProjection* is executed. *algoProjection* first finds all frequent pairs (timestamp, item) that verify the constraints C1, C2 and *minsup*. For example, for ISDB| (0,a), the frequent pair (0,b) and (2,a) are found. These pairs are concatenated to (0,a) to obtain sequences <(0,a), (0,b)> and <(0,a), (2,a)>, respectively. Because these sequences respect C3 and C4, they are added to the set R of sequences found. Then, the projected database ISDB|<(0,a), (0,b)> and ISDB|<(0,a), (2,a)> are calculated. Because these databases contain more than *minsup* sequences, *algoProjection* is executed again. After completing the execution of the algorithm, the set of sequences R contains <(0,a)>, <(0,a), (0,b)>, <(0,a), (2,a)>, <(0,b)> and <(0,c)>.

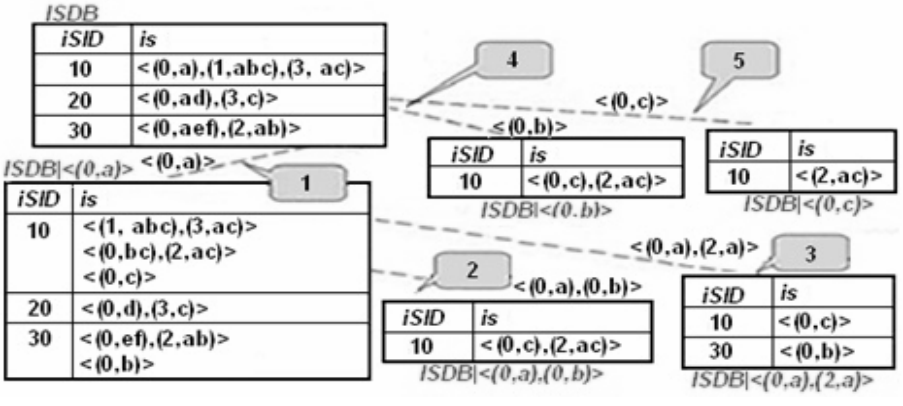


Fig. 3. An application of the Hirate-Yamana Algorithm (adapted from [10])

5 Extending SPM with Automatic Clustering of Valued Actions

A second limitation that we encountered when we applied PrefixSpan to extract a problem space is that it relies on a finite set of actions. As a consequence, if some actions were designed to have parameters or values, they have to be defined as one or more distinct actions. For example, in our first experiment in RomanTutor, we categorized the joint rotations as small, medium and big, which correspond respectively to 0

to 60, 60 to 100, and more than 140 degrees. The disadvantage with this categorization is that it is fixed. In order to have dynamic categories of actions, we extended the Hirate-Yamana algorithm to perform an automatic clustering of valued actions.

We propose to define a valued sequence database as a time-extended sequence database, where sequences can contain valued actions. A valued action *a{value}*, is defined as an action *a* that has a value *value*. In this work, we consider that a value is an integer. For example, the sequence <(0,a{2}), (1,b), (2,bc{4})> contains the valued action *a*, and *b* with values 2 and 4, respectively. The left part of figure 4 shows an example of a sequence database containing valued actions.

ID	Time-extended sequences
1	<(0,a{2}), (1,bc{4})>
2	<(0,a{2}), (1,c{5}))>
3	<(0,a{5}), (1,c{6}))>
4	< (0,f), (1,a{6}))>
5	<(0, f b{3}), (1,e) , (2,f))>
6	<(0,b{2}), (1,d))>

→

Mined valued seq. patterns	Supp.
<(0,a{2})>	33 %
<(0,a{5})>	33 %
<(0,a{2}), (1, c{5})>	33 %
<(0,c{5})>	50 %
<(0,f)>	33 %
…	…

Fig. 4. Applying Hirate-Yamana with Automatic Clustering of Valued Actions

To mine patterns from a valued database, we added a special treatment for valued actions. We modified the action/pair counting of the Hirate-Yamana algorithm to note the values of the action being counted, and their sequence ids. We modified the database projection operation ISDB|(t,i) so that it can be called with a valued action *i* and a set of values V={v_1, v_2…, v_n}. If the support of *i* in ISDB is higher or equals to 2 * *minsup*, the database projection operation calls the K-Means algorithm [15] to find clusters. The K-Means algorithm takes as parameter K, a number of clusters to be created, and the set of values V to be clustered. K-Means first creates K random clusters. Then it iteratively assigns each value from V to the cluster with the nearest median value until all clusters remain the same for two successive iterations. In the database projection operation, K-Means is executed several times starting with K=2, and incrementing K until the number of frequent clusters found (with size >= *minsup*) does not increase. This larger set of frequent clusters is kept. Then, the sequences of ISDB|(t,i) are separated into one or more databases according to these clusters. Afterward, *algoProjection* is called for each of these databases with size equal or greater than *minsup*.

Moreover, if ISDB|(t,i) is called from *algoHirate* and *n* clusters are found, instead of just adding <(o,{i})> to the set R of sequences found, <(0, i{v_{x1}})>, <(0, i{v_{x2}})> … <(0, i{v_{xn}})> are added, where v_{x1}, v_{x2}, .. v_{xn} are the median value of each cluster. Similarly, we have adapted *algoProjection* so that sequences are grown by executing *Concatenate* with (t, i{v_{x1}}), (t, i{v_{x2}}) … (t, i{v_{xn}}, instead of only <(t,{i})>.

The right part of figure 4 shows some sequences obtained from the execution of the modified algorithm with a *minsup* of 32 % (2 sequences) on the valued sequence database depicted in the left part of figure 4. From this example, we can see that the action “a” is frequent (with a support of 4) and that two clusters were dynamically created from the set of values {2, 2, 5, 6} associated to “a”. The first cluster contained the values 2 and 2 with a median value of 2 and the second one contained 5 and 6

with a median value of 5. This resulted in frequent sequences containing "a{2}"and some other sequences containing "a{5}".

The advantage of the modified algorithm over creating fixed action categories is that actions are automatically grouped together based on their similarity and that the median value is kept as an indication of the values grouped. Note that more information could be kept such as the minimum and maximum values for each cluster, and that a different clustering algorithm could be used.

A test with logs from RomanTutor permitted extracting sequential patterns containing valued actions. One such pattern indicates that learners performed a rotation of joint EP with a median value of 15º, followed by selecting camera 6. Another pattern found consists of applying a rotation of joint EP with a median value of 53º followed by the selection of the camera 4. In this case, the dynamic categorization enhanced the quality of the extracted patterns, since otherwise both patterns would be considered as starting with a "small rotation" of the joint EP.

6 Extending the Hirate-Yamana Algorithm to Mine the Compact Representation of Closed Sequences

A third limitation of the Hirate-Yamana algorithm and of the SPM algorithms such as Prefixspan is that among the mined sequences, there can be many redundant sequences. For example, the Hirate-Yamana algorithm could find the three frequent sequences <(0,a)> , <(0,a), (0,b)> and <(0,a), (0,b), (0,c)> in a database. In a tutoring context where we want to extract a task model, only the frequent closed or maximal sequences could be kept, as we are interested by the longer sequences, and this would allow reducing the number of sequences to consider by tutoring services.

"Closed sequences" are sequences that are not contained in another sequence having the same support. A closed pattern induces an equivalence class of pattern sharing the same closure, i.e. all the patterns belonging to the equivalence class are verified by exactly the same set of plans. Those patterns are partially ordered, e.g. considering the inclusion relation. The smallest elements in the equivalence class are called minimal generators, and the unique maximal element is called the closed pattern. On the other hand, "maximal sequences" are sequences that are not contained in any other sequence. In this work, we extend our algorithm to mine closed sequences instead of maximal sequences, since closed sequences are a lossless compact representation of the set of frequent sequences. In other words, the set of closed frequent sequences allows reconstituting the set of all frequent sequences and their support [16] (no information is loss).

We have extended our modified Hirate-Yamana algorithm to find only closed sequences. To achieve this, we have integrated the BI-Directional Extension closure (BIDE) checking of the BIDE+ algorithm [16], which was proposed as an extension of the PrefixSpan algorithm, and permits checking if a sequence is closed without having to maintain a set of closed sequences candidates (as many closed pattern mining algorithm do). The BIDE scheme basically checks if a pattern is closed by checking in the original sequences that contains the pattern if there exist one action with the same support that could extend the pattern. We have also implemented the BackScan pruning of the BIDE+ algorithm that allows stopping growing some sequences that

are guaranteed to not produce any closed sequences (see [16] for further information). The BackScan pruning has the advantage of often increasing the time performance over regular SPM algorithms such as PrefixSpan [16]. Because of limited space, the reader is invited to refer to [16] for more details on the BIDE algorithm.

7 Extending Sequential Pattern Mining with Context Information

A fourth limitation that we encountered when applying the PrefixSpan algorithm is that it does not consider the context of each sequence. In a tutoring system context, it would be useful, for instance, to annotate sequences with success information and the expertise level of a user and to mine patterns containing this information. Our solution to this issue is to add dimensional information to sequences. Pinto et al. [12] originally proposed Multi-dimensional Sequential Pattern Mining (MDSPM), as an extension to SPM. A Multidimensional-Database (MD-Database) is defined as a sequence database having a set of dimensions $D=\{D_1, D_2,\ldots D_n\}$. Each sequence of a MD-Database (an MD-Sequence) possesses a symbolic value for each dimension. This set of value is called an MD-Pattern and is noted $\{d_1, d_2\ldots d_n\}$. For example, consider the MD-Database depicted in the left part of figure 5. The MD-Sequence 1 has the MD-Pattern {"true", "novice"} for the dimensions "success" and "expertise level". The symbol "*", which means any values, can also be used in an MD-Pattern. This symbol subsumes all other dimension values. An MD-Pattern $P_x=\{d_{x1}, d_{x2}\ldots d_{xn}\}$ is said to be contained in another MD-Pattern $P_y=\{d_{y1}, d_{y2}\ldots d_{ym}\}$ if there exists integers $1 \leq i_1 < i_2 < \ldots i_n \leq m$ such that $d_{x1} \subseteq d_{y1}, d_{x2} \subseteq d_{y2}, \ldots, d_{xn} \subseteq d_{yn}$. The problem of MDSPM is to find all MD-Sequence appearing in a database with a support higher than *minsup*. Figure 5 shows an MD-Database with time information and some patterns that can be extracted from it, with a *minsup* of 2 sequences.

An MD-Database		
ID	**Dimensions**	**Sequences**
1	true, novice	<(0,a),(1,bc)>
2	true, expert	<(0,d) >
3	false, novice	<(0,a),(1,bc)>
4	false, interm.	<(0,a),(1,c), (2,d)>
5	true, novice	<(0,d), (1,c)>
6	true, expert	<(0,c), (1,d)

→

Mined MD-Sequences	
Dimensions	**Sequences**
*, novice,	<(0,a)>
*, *	<(0,a)>
*, novice	<(0,a), (1,b)>
true, *	<(0,d)>
true, novice	<(0,c)>
true, expert	<(0,d)>

Fig. 5. An Example of SPM with Dimensions and Time Information

Pinto et al. [17] proposed three algorithms for MDSPM. The first one cannot be applied in combination with the Hirate-Yamana algorithm, as it required embedding dimensions as additional actions in sequences. The two other algorithms, SeqDim and DimSeq are based on the idea that the problem of MDSPM can be broken in two steps: finding sequential patterns with an algorithm such as PrefixSpan, and finding MD-Patterns with an itemset mining algorithm such as Apriori [18, 19]. The first algorithm, SeqDim is executed as follow. First, frequent sequences are found by SPM. Then, for each sequence, the containing MD-Sequences are used to mine frequent MD-Patterns

which are then combined with the sequence to form MD-Sequence(s). Alternatively, the second algorithm, DimSeq, first perform a search for frequent MD-Patterns. Then, for each pattern, the containing MD-Sequences are used to mine frequent sequences, which are then combined with the pattern to form MD-Sequence(s).

In our implementation, we chose SeqDim and integrated it with our extended Hirate-Yamana algorithm. For MD-Patterns mining, we applied the AprioriClose algorithm [19], but any itemset mining algorithm such as Apriori [17] can be used. We chose AprioriClose as it allows mining the set of closed MD-Patterns, and thus to eliminate some redundancy among the mined sequences. It is important to note that combining closed MD-Patterns mining and closed SPM does not results in closed MDSPM [20]. In future work we plan to adapt our algorithm as suggested by [20] to achieve closed MDSPM, and remove more redundancy.

Applying our modified algorithm with MDSPM in RomanTutor showed to be useful as it allowed to successfully identify patterns common to all expertise levels that lead to failure ("*, failure"), for example. Currently, we have encoded two dimensions: expertise level and success. But additional dimensions can be easily added. In future work, we plan to encode skills involved as dimensional information (each skill could be encoded as a dimension). This will allow computing a subset of skills that characterize a pattern by finding common skills demonstrated by users who used that pattern. This will allow diagnosing missing and misunderstanding skill for users who demonstrated a pattern.

8 Dividing Long Problems into Sub-problems

The last improvement that we made is to how we apply the algorithm to extract a partial problem space. Originally, we mined frequent patterns from sequences of user actions for a whole problem-solving exercise. But, we noticed that in general, after more than 6 actions performed by a learner, it becomes hard for the system to tell which pattern the learner is doing. For this reason, we added the definition of "problem states". For example, in the RomanTutor, where an exercise consists of moving a robotic arm to attain a specific arm configuration, the 3D space was divided into 3D cubes, and the problem state at a given moment is defined as the set of cubes containing the arm joints. An exercise is then viewed as going from a problem state P_1 to a problem state P_F. For each attempt at solving the exercise, we log (1) the sequence of problem states visited by the learner $A=\{P_1, P_2 \ldots P_n\}$ and (2) the list of actions performed by the learner to go from each problem state to the next visited problem state (P_1 to P_2, P_2 to P_3, ... P_{n-1} to P_n). After many users performed the same exercise, we extract sequential patterns from (1) the sequences of problems states visited, and (2) from the sequences of actions performed for going from a problem state to another.

Dividing long problems into sub-problems allow a better guidance of the learner, because at any moment, only the patterns starting from the current problem state have to be considered.

We describe next how the main tutoring services are implemented. To recognize a learner's plan, the system proceeds as follow. The first action of the learner is compared with the first action of each pattern for the current problem state. The system discards the patterns that do not match. Each time the learner makes an action, the

system compares the actions done so far by the learner with the remaining patterns. When the problem-state changes, the system considers the set of patterns associated to the new problem state. If at any given moment a user action does not match with any patterns, the algorithm ignores the last user action or the current action to match for each pattern. This makes the plan recognizing algorithm more flexible and has shown to improve its effectiveness. At a more coarse grain level, a tracking of the problem states visited by the learners is achieved similarly as the tracking for actions.

One utility of the plan recognizing algorithm for actions/problem states is to assess the expertise level of the learner (novice, intermediate or expert) by looking at the patterns applied. The plan recognizing algorithm also allows suggesting to the learner the possible actions from the current state. In RomanTutor, this functionality is triggered when the student selects "What should I do next?" in the interface menu. In this case, the tutoring service selects the action among the set of patterns that has the highest relative support and that is the most appropriate for the estimated expertise level of the learner. When no actions can be identified, RomanTutor relies on a path-planner [6] to generate an approximate path to the goal.

9 A Preliminary Experiment

We conducted a preliminary experiment in RomanTutor. We have set up two scenarios consisting each of moving a load with the Canadarm2 robotic arm to one of the two cubes (figure 6). A total of 12 users (a mix of novices, intermediates and experts) have been invited to execute these scenarios using the CanadarmII robot simulator. The number of primitive actions that have been retained is 112 (some have been redefined from the original 155 to have values, thus reducing the number of different possible actions). The expertise levels and success/failure information was added manually to sequences. From this data set, we extracted 558 sequential patterns for problem-states and actions with the extended SPM algorithm. These patterns were then used as input by the tutoring services of RomanTutor. In a subsequent work session, we asked the users to compare the tutoring services offered in our first experiment with RomanTutor with those offered with the newly extracted knowledge base. On the whole, users preferred the newer version, as the hints offered were generally more precise and more appropriate, and help could be provided in more situations. It was also observed that the system more often correctly inferred the estimated expertise level of learners by using the dimensional information.

Figure 7 illustrates a hint message given to a learner upon request during scenario 1. The guiding tutoring service selected the pattern that has the highest support value, matches the last student actions and problem-state, is marked "success" and corresponds with the estimated expertise level of the learner. The given hint is to decrease the rotation value of the joint "EP", increase the rotation value of joint "WY", and finally to select camera "CP2" on "Monitor1". The values on the right column indicate the values associated to the action. In this context, the values "2" and "3" means to rotate the joints 20 º and 30 º, respectively (1 unit equals 10º). By default, three steps are showed to the learners in the hint window depicted in figure 7. However, the learner can click on the "More" button to ask for more steps or click on the "another possibility" button to ask for an alternative.

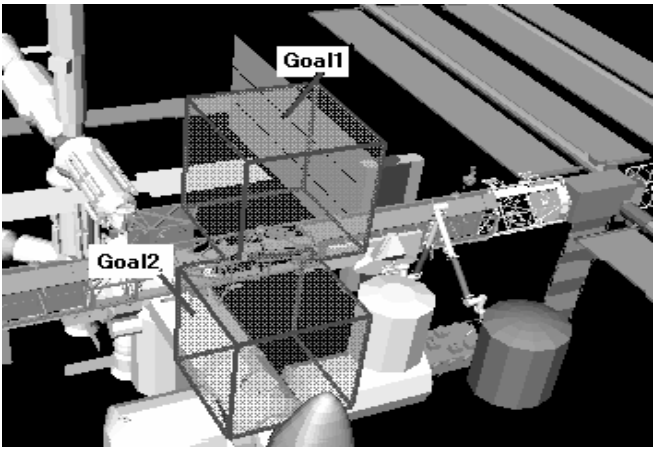


Fig. 6. The two manipulation scenarios

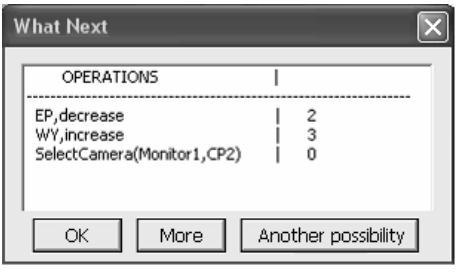


Fig. 7. A hint generated by the guiding tutoring service

10 Conclusion

Following our first experiment of applying a SPM based framework [5] in RomanTutor, we reported in this paper several limitations of SPM for learning procedural knowledge associated to a task, and proposed a new SPM framework to overcome these limitations. The SPM algorithm combines (1) time intervals, (2) closed sequential pattern mining, (3) multi-dimensional pattern mining and (4) the automatic clustering of valued actions. We also suggested dividing problems into problem states to enhance the relevance of the tutoring services.

The framework was used to extract a problem space, and support tutoring services in RomanTutor. Since the framework proposed in this paper and its inputs and outputs are domain independent, the framework can be potentially applied to any ill-defined procedural domains where the problem can be stated in the same way.

For future work, we are first working on including skills as dimensional information, and on conducting an experiment with a larger group of learners. We are also planning to develop new tutoring services to exploit the problem space.

Finally, we plan to use association rules mining as in our previous work, to find associations between patterns over a whole problem-solving exercise [5]. This could

improve the effectiveness of the tutoring services, as it would be complementary to dividing the problem into problem states. For example, if a learner performed a pattern p, an association rule could indicate that the learner has a higher probability of applying another pattern q later during the exercise than another pattern r that is available for the same problem state.

Acknowledgment. Our thanks go to the FQRNT and NSERC for their logistic and financial support. The authors would like to thank Severin Vigot and Mikael Watrelot for integrating the framework in RomanTutor, and Khaled Belghith, Daniel Dubois, Usef Faghihi, Mohamed Gaha and the other members of the GDAC/PLANIART teams who participated in the development of RomanTutor.

References

[1] Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.: The Cognitive Tutor Authoring Tools (CTAT): Preliminary evaluation of efficiency gains. In: Proc. of ITS 2006, pp. 61–70 (2006)

[2] Simon, H.A.: Information-processing theory of human problem solving. In: Estes, W.K. (ed.) Handbook of learning and cognitive processes. Human information, vol. 5 (1978)

[3] Lynch, C., Ashley, K., Aleven, V., Pinkwart, N.: Defining Ill-Defined Domains; A literature survey. In: Proc. of the Intelligent Tutoring Systems for Ill-Defined Domains Workshop, ITS 2006, pp. 1–10 (2006)

[4] Mitrovic, A., Mayo, M., Suraweera, P., Martin, B.: Contraint-based tutors: a success story. In: Proc. of the Industrial & Engineering Application of Artificial Intelligence & Expert Systems, pp. 931–940 (2001)

[5] Nkambou, R., Mephu Nguifo, E., Fournier-Viger, P.: Using Knowledge Discovery Techniques to Support Tutoring in an Ill-Defined Domain. In: Proc. 9th Int. Conf. Intelligent Tutoring Systems (ITS 2008) (2008)

[6] Kabanza, F., Nkambou, R., Belghith, K.: Path-planning for Autonomous Training on Robot Manipulators in Space. In: Proc. of IJCAI 2005 (2005)

[7] Fournier-Viger, P., Nkambou, R., Mayers., A.: A Framework for Evaluating Semantic Knowledge in Problem-Solving-Based Intelligent Tutoring Systems. In: Proc. of FLAIRS 2008, pp. 409–414. AAAI press, Menlo Park (2008)

[8] McLaren, B., et al.: Bootstrapping Novice Data: Semi-Automated Tutor Authoring Using Student Log Files. In: Proc. of the Workshop on Analyzing Student-Tutor Logs. ITS 2004 (2004)

[9] Jarivs, M., Nuzzo-Jones, G., Heffernan, N.T.: Applying Machine Learning Techniques to Rule Generation in Intelligent Tutoring Systems. In: Proc. of ITS 2006, pp. 541–553 (2006)

[10] Blessing, S.B.: A Programming by Demonstration Authoring Tool for Model-Tracing Tutors. In: Murray, T., Blessing, S., Ainsworth, S. (eds.) Authoring Tools for Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive and Intelligent Educational Software, pp. 93–119. Kluwer Academic Publ., Dordrecht (2003)

[11] Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: Proc. Int. Conf. on Data Engineering, pp. 3–14 (1995)

[12] Pei, J., Han, J., et al.: Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. IEEE Trans. Knowledge and Data Engineering 16(10), 1–17 (2004)

[13] Zaki, M.J.: SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning Journal 42(1-2), 31–60 (2001)
[14] Hirate, Y., Yamana, H.: Generalized Sequential Pattern Mining with Item Intervals. Journal of Computers 1(3), 51–60 (2006)
[15] MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proc. 5th Berkeley Symposium on Mathematic Statistics and Probability, pp. 281–297 (1967)
[16] Wang, J., Han, J., Li, C.: Frequent Closed Sequence Mining without Candidate Maintenance. IEEE Transactions on Knowledge and Data Engineering 19(8), 1042–1056 (2007)
[17] Pinto, H., et al.: Multi-Dimensional Sequential Pattern Mining. In: Proc. Int. Conf. Information and Knowledge Management (CIKM 2001), pp. 81–88 (2001)
[18] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proc. 20th Int. Conf. Very Large Data Bases, VLDB, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
[19] Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Closed set based discovery of small covers for association rules. In: Proc. 15emes Journees Bases de Donnees Avancees, pp. 361–381 (1999)
[20] Songram, P., Boonjing, V., Intakosum, S.: Closed Multidimensional Sequential Pattern Mining. In: Proc. 3rd Int. Conf. Information Technology: New Generations, pp. 512–517 (2006)

Enhancing Teams with PANCHO to Improve the Training Experience

Raúl A. Aguilar[1], Angélica de Antonio[2], and Ricardo Imbert[2]

[1] Universidad Autónoma de Yucatán, Mathematics School
Periférico Norte Tablaje 13615, A.P. 172, Cordemex, C.P. 97110, Mérida, Mexico
avera@uady.mx
[2] Universidad Politécnica de Madrid, Computer Science School
Campus Montegancedo, 28660, Boadilla del Monte, Madrid, Spain
{angelica,rimbert}@fi.upm.es

Abstract. Here we present an automated mechanism used by a human coach in the decision making process to enhance teams replacing a human member with an Intelligent Virtual Agent (IVA). The IVA called PANCHO (Pedagogical AgeNt to support Collaborative Human grOups) will join the human group with the aim of improving the performance of the team —inspired by the Team Roles Theory— and providing scaffolding to the trainees during the task execution. PANCHO´s architecture —an emotional architecture— is also described briefly.

Keywords: Intelligent Virtual Agent, Team Roles, Team Training.

1 Introduction

Intelligent Virtual Agents (IVAs) have characteristics that allow increasing the computer's ability to engage and motivate groups of trainees (Teams) along their learning or training process. The IVA cohabits the virtual world with human trainees and collaborates with them on training scenarios [1]. Within Collaborative Virtual Environments (CVEs) for Training, an IVA can carry out two important functions: first, the agent can substitute missing team members, allowing trainees to practice a team task; and, second, they can play the role of a coach to improve the trainee and/or team learning and behavior by providing advice, help, feedback, etc.

Our interest on Intelligent Virtual Environments for Training (IVETs) has led us to propose a Team Training Strategy — for small groups— in which the human team to be trained follows an iterative process of self-assessment about the execution of a proposed task.

The strategy proposes the replacement of a human team member for an IVA called PANCHO with the aim of improving the performance of the team (*Teamwork*) and providing scaffolding to the trainees (*Taskwork*) during the practice task [2].

PANCHO (Pedagogical AgeNt to support Collaborative Human grOups) is an Intelligent Virtual Agent that we have designed —using an emotion based Architecture— to generate the particular behaviors of Team Roles as defined by Belbin [3][4]; the goal is to join to human group an IVA with a selected team role —according to a

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 779–788, 2008.

team model previously obtained— with the intention of improving the performance of the team (Teamwork) and providing scaffolding to the trainees (Taskwork). According to the defined Team Training Strategy (TTS), the team (PANCHO included) will exercise in an IVET using like shared knowledge the co-constructed execution plan obtained in a previous stage.

In the following section we briefly describe the Team Training Strategy used as a reference in this work; the third section describes the main concepts related to the team roles and the automated mechanism —incorporated into a CVE— that we have designed to select the human team member to replace, as well as to select the team role that PANCHO should play in the team; finally, the general architecture used to create PANCHO is briefly described.

2 The Team Training Strategy (TTS)

The TTS consists of five interrelated stages through which the human team to be trained follows an iterative process of self-assessment about the execution of a plan to perform a proposed task (see fig. 1). The interaction level between trainees is expected to keep on increasing while the team makes progress through the strategy.

The *Integration* stage has the purpose of integrating the human team with the support of a Collaborative Virtual Environment, as well as providing the trainees with a first mental schema of the plan to be executed for a predefined task. In the following stage (*Planning*) the team members, in a collaborative virtual meeting, co-construct an execution plan for the task. In the third stage (*Execution*) the team uses an IVET to execute the planned activities according to their assigned roles. In the fourth stage (*Evaluation*) the team members will have to evaluate their previous performance and must identify both individual and group errors with the purpose of avoiding them in a future execution stage. Finally, in the last stage (*Improvement*) the team members co-construct a new plan for the task using as a baseline the experience acquired during the iterative execution and evaluation of the initial plan.

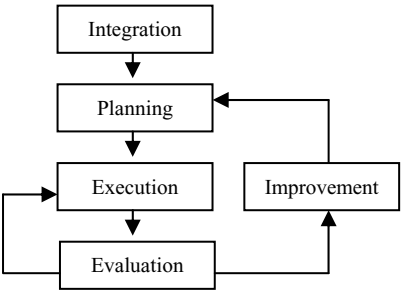


Fig. 1. The Team Training Strategy (TTS)

The iterative process followed by the team promotes activities and situations which are expected to trigger learning mechanisms between the team members allowing a better performance during the teamwork. The potential domains for the application of the TTS are the control and maintenance of industrial plants, operation and maintenance

of industrial equipment, coordinated manipulation of vehicles and, especially, human rescue teams for disaster situations.

3 The Theory of Roles for Teamwork

In our days, organizations can adopt several structures or patterns of information and control relations; these structures define the jobs and tasks required for each functional role on the team; however, for certain kind of tasks flat structures without authority hierarchies have demonstrated more efficiency; in these cases, the non-functional roles or team roles acquire major importance. Many researchers and consultants claim to have identified sets of team roles, which, when fully represented among team members, are said to improve the team's performance.

3.1 The Team Roles

Belbin´s work [3][4] is the most popular categorization of team roles; the transcendence of this work possibly takes root in that it isn't just a classification, but it also offers principles about teamwork. Table 1 shows the nine Team Roles (TR) proposed by Belbin, which are classified in accordance to his orientation in three kinds (KRole): Action oriented (A), People oriented (P), and Cerebral (C).

Table 1. Belbin's Team Roles

Role	Description
A. Sharper (SH)	Dynamic, challenging. Has drive and courage to overcome obstacles.
A. Implementer (IM)	Disciplines, reliable, conservative. Turns ideas into practical actions.
A. Completer Finisher (CF)	Painstaking, conscientious, anxious. Searches out errors and omissions, delivers on time.
P. Coordinator (CO)	Mature, confident, a good chairperson. Clarifies goals, promotes decision making.
P. Team Worker (TW)	Co-operative, mild, perceptive, diplomatic. Listens, builds, averts friction.
P. Resource Investigator (RI)	Extrovert, enthusiastic. Explores opportunities. Develops contacts.
C. Plant (PL)	Creative, imaginative, unorthodox. Solves difficult problems.
C. Monitor-Evaluator (ME)	Sober, strategic, discerning. Sees all options.
C Specialist (SP)	Single-minded, self starting. Dedicated. Provides knowledge and skills in rare supply.

3.2 A Self-perception Inventory

Belbin states that a Team Role could be defined as a tendency to behave, contribute and interrelate with others at work in a certain distinctive way. As part of his research on team roles, Belbin also designed an instrument for measuring the team roles called

Belbin´s Self-Perception Inventory (SPI). It is a questionnaire with seven sections in which the trainee distributes ten points among eight statements, trying to give the best description about his behavior.

We have automated the analysis of Belbin´s SPI (see data in figure 2) for identifying the Best Perceived Role (BPR) of each trainee (T_x).

$$TR(T_x) = BPR(T_x)$$

The figure 2 shows an individual report resulting from Belbin´s SPI in which *Implementer* is identified as the BPR —with a value of affinity of 22% (Val(BPR(T_x)) = 22%)— for the Trainee: *Alumno-007*.

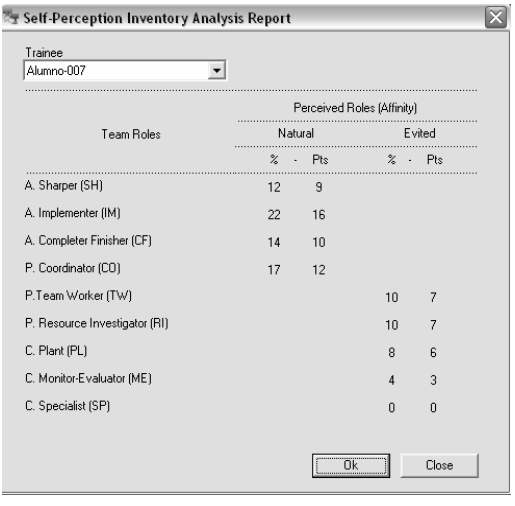


Fig. 2. Individual Report of the SPI

4 An Automated Mechanism for Enhance Teams

Because our TTS is oriented to small groups, we have selected groups integrated by three members; according to Bean in [5] this is an optimal group size for the workgroup. We think that the individual specific weight into the group interaction with this group size could result significant and we propose to replace a team member by PANCHO in the execution stage with the intention of improving the training experience.

In order to perform this substitution we have developed an automated mechanism to give answer to the two following questions:

- Who's the member of the human group who might be replaced? and,
- What's the team role that PANCHO should play?

4.1 Whom to Replace?

Using the team roles theory as a reference, we defined two criteria to answer this first question:

1. Compatibilities between the TRs identified for the members of the group.
2. Affinity of the BPR.

For the compatibility analysis between the different team roles that integrate the Team (HTeam) for each member of the human group (HGroup)

$HTeam = \{ TR(T_x), TR(T_y), TR(T_z) \}$ $HGroup = \{ T_x, T_y\ T_z \}$

With $x \neq y$, $x \neq z$ and $y \neq z$

We will consider $\forall\ T_x, T_y\ T_z \in HGroup$ with $x \neq y$, $x \neq z$ and $y \neq z$:

If $(TR(T_x) = SH) \wedge (TR(T_y) \in \{CO,TW\})$ → Then (T_x,T_y) = Incompatible.

If $(TR(T_x) = IM) \wedge (TR(T_y) \in \{IM,PL\})$ → Then (T_x,T_y) = Incompatible.

If $(TR(T_x) = CF) \wedge (TR(T_y) \in \{RI,ME\})$ → Then (T_x,T_y) = Incompatible.

If $(TR(T_x) = PL) \wedge (TR(T_y) \in \{ME,PL\})$ → Then (T_x,T_y) = Incompatible.

If $(TR(T_x) = ME) \wedge (TR(T_y) \in \{CF,ME,PL,SP\})$ → Then (T_x,T_y) = Incompatible.

If $(TR(T_x) = SP) \wedge (TR(T_y) \in \{PL,RI\})$ → Then (T_x,T_y) = Incompatible.

If $(TR(T_x) = CO) \wedge (TR(T_y) = SH)$ → Then (T_x,T_y) = Incompatible.

If $(TR(T_x) = RI) \wedge (TR(T_y) \in \{CF,SP\})$ → Then (T_x,T_y) = Incompatible.

If $(TR(T_x) = TW) \wedge (TR(T_y) = SH)$ → Then (T_x,T_y) = Incompatible.

Then, the recommendation will be:

If (T_x,T_y) = Incomp. $\wedge$ (T_x,T_z) = Incomp. $\wedge$ (T_y,T_z) = Incomp.

→ The HTeam can't be improved by replacing just one member.

If (T_x,T_y) = Incomp. $\wedge$ (T_x,T_z) = Incomp. $\wedge$ $(T_y,T_z) \neq$ Incomp.

→ Replace T_x.

If (T_x,T_y) = Incomp. $\wedge$ $(T_x,T_z) \neq$ Incomp. $\wedge$ $(T_y,T_z) \neq$ Incomp.

→ Replace T_n. with $Val(BPR(T_n)) = MIN\ (Val(BPR(T_x)), Val(BPR(T_y)))$.

If $(T_x,T_y) \neq$ Incomp. $\wedge$ $(T_x,T_z) \neq$ Incomp. $\wedge$ $(T_y,T_z) \neq$ Incomp.

→ Replace T_n. with $Val(BPR(T_n)) = MIN\ (Val(BPR(T_x)), Val(BPR(T_z)), Val(BPR(T_y)))$.

4.2 What Team Role Should PANCHO Play?

On the other hand, to answer this second question, the criteria used to select the team role most apt to be assumed by PANCHO are:

1. The task type that the team will have to realize.
2. The balance between the kinds of roles existing in the team (KRole).
3. Compatibilities with the other two teammates.

To establish the Task Influence (TI) in the selection of the kind of role for PANCHO (KRole(Pancho)) we have used the classification proposed by McGrath in [5]. This framework proposes a classification of tasks into four general processes: *Generate*, *Choose*, *Negotiate*, and *Execute*. Nevertheless, because the proposed task

might include activities of the four proposed types, we leave to the human tutor the possibility of assigning a weight for each of these, with the option to assign a 100% to only one type of task, or to distribute it between all types.

According to the type of task, and considering the descriptions proposed by Belbin for each kind of team role, we have defined the influence of the three kinds of role using a reasoning similar to the reported in [6] about the influence in the interaction level by the type of member of the group; in this way, we will assign a weight to the most significant role type equivalent to that of the rest of the role types (50%), and the influence of the second most important will be at least double (34%) than the least important (16%) (see table 2).

Table 2. Influence relations Task-Role

			Kind of Role Influence		
Task Type (TT)	Weight (TT_i)		Action (A)	People (P)	Cerebral (C)
TT_1 = Generate	Weight (TT_1)	*	(.16)	(.34)	(.50)
TT_2 = Choose	Weight (TT_2)	*	(.16)	(.50)	(.34)
TT_3 = Negotiate	Weight (TT_3)	*	(.16)	(.50)	(.34)
TT_4 = Execute	Weight (TT_4)	*	(.50)	(.34)	(.16)
Task Influence (TI)			KChoice[A]	KChoice[P]	KChoice[C]

Where for instance:

$$\text{KChoice}[A] = \sum_{i=1}^{4} \text{Weight}(TTi) * \text{Influence}(A,i)$$

To select the kind of role that PANCHO must play, we consider $\forall\ T_x, T_y \in$ HGroup with $x \neq y$, and considering that the ideal team should have a balance of the role types:

$\text{RealTeam} = \{\text{KRole}(T_x), \text{KRole}(T_y)\}$ $\quad\quad$ $\text{IdealTeam} = \{A, P, C\}$

If $\text{KRole}(T_x) \neq \text{KRole}(T_y)$ → $\text{KRole(Pancho)} \in (\text{IdealTeam} - \text{RealTeam})$.

If $\text{KRole}(T_x) = \text{KRole}(T_y)$ → $\text{KRole(Pancho)} = p \mid \text{KChoice}[p] > \text{KChoice}[q]$ where $p \neq q$, $p \neq \text{KRole}(T_x)$ and $q \neq \text{KRole}(T_x)$.

As soon as the kind of role is selected, we can suggest, using the incompatibilities analysis between the other two members of the team, the set of team roles that PANCHO might assume. As an example, if PANCHOS´s kind of role turned out to be People Oriented, then:

If $(\text{KRole(Pancho)} = P) \wedge (\text{TR}(T_x) = SH \vee \text{TR}(T_y) = SH)$ → $\text{TR(Pancho)} = RI$.

If $(\text{KRole(Pancho)} = P) \wedge (\text{TR}(T_x) \in \{CF,SP\} \vee \text{TR}(T_y) \in \{CF,SP\})$

→ $\text{TR(Pancho)} \in \{CO,TW\}$

If $(\text{KRole(Pancho)} = P) \wedge (\text{TR}(T_x) \notin \{SH,CF,SP\}) \wedge (\text{TR}(T_y) \notin \{SH,CF,SP\})$

→ $\text{TR(Pancho)} \in \{CO,RI,TW\}$

A similar analysis has been developed with the other two kinds of roles (A, C). The figure 3 shows the reports obtained for the analysis of a human group in which our mechanism has proposed to replace the trainee with the *Sharper* role.

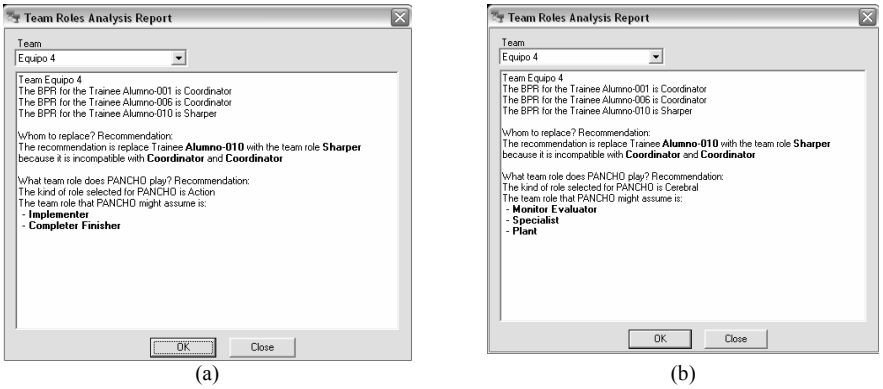


Fig. 3. Team Report for the team roles analysis

As an illustration of how Task Type influences the selection, for the report of to the figure 3a we have assigned the values: *Generate: 20%* and *Execute: 80%* and the recommendation is an Action oriented role; in the other case (figure 3b) we have assigned the values: *Generate: 80%* and *Execute: 20%* and the recommendation is a Cerebral role.

5 The PANCHOs Architecture

Because the team roles that PANCHO must perform have very particular behaviors, we have selected a cognitive architecture for agents with emotionally influenced behaviors —called COGNITIVA— to realize those roles [7].

5.1 A Brief Review of COGNITIVA

COGNITIVA is a multi-layered emotional architecture for the development of IVAs. It presents three layers to deal with different types of behaviors: a reactive layer, to provide immediate responses to events perceived from the environment; a deliberative layer, to generate goal driven behaviors; and a social layer, to manage behaviors that take into account interactions with other agents.

COGNITIVA proposes that an IVA should have beliefs about the most probable state of the environment, including beliefs about: places, objects, individuals and about itself (the so called Personal Model). For their management, COGNITIVA proposes a taxonomy of beliefs as follows:

- *Defining Characteristics (DC)*. Which describe the general traits of the places, objects, and people.
- *Transitory States (TS*). Characteristics whose values represent the current state of the element in the environment.

- *Attitudes (A)*. Parameters that determine the behavior of an IVA towards other environmental elements.

COGNITIVA also proposes a progressive specification process, which allows applying it in a variety of problems and contexts. As a general outline, this process consists of two phases of specification.

- *Functional Specification*. In this phase the developer details the particular structure and format of every component of the architecture, specifying also the design of all the functions proposed by the architecture.
- *Contextual Specification*. This phase deals with the particularization of the concrete values needed for every architectural element, according to the application context. In this phase, for instance, we decided that we needed an agent of the kind "team role", whose behavior will be influenced by properly calibrated personality factors.

5.2 Extending the Specification Process for PANCHO

COGNITIVA has proposed as a possible *Functional Specification*, a fuzzy logic representation for beliefs and their relationships; in PANCHO we have used this specification with slight modifications.

Because the general behaviors of team roles are quite independent of the concrete training domain, in our work we have split the *Contextual Specification* (second phase) into two new phases:

- *Group Dynamics Specification*. In this phase the IVA´s Personal Model specification for a certain team role, the kind of beliefs about the teammates that it is capable of supporting, and the social-emotional actions at teamwork according to the team role, are stated.
- *Training Domain Specification*. The second phase deals with the specification of beliefs about the places and objects, and the possible instrumental actions in the application context.

5.3 The Beliefs in PANCHO

As part of the *Group Dynamics Specification*, two groups of beliefs for an agent PANCHO were defined: beliefs about himself, and beliefs about the teammates. Using the taxonomy defined in COGNITIVA, the beliefs about himself will be represented as follows:

$$Bel(P) = DC(P) \cup TS(P) \cup A(P) \quad (1)$$

We are include into the *DC(P)* (see 2) beliefs about the identifiers of the trainee and team, about the team role (*TR*) and kind of team role (*KR*) —using Belbin´s Theory— that PANCHO carries out, and beliefs about the personality traits (*P*).

$$DC(P) = IdTrainee(P) \cup IdTeam(P) \cup TR(P) \cup KR(P) \cup P(P) \quad (2)$$

To generate the desired behaviors we have specified the personality traits carefully —using the Big Five Model [8]. A group of heuristics was used to define the values for each of the traits, using as reference the work of Lindgren [9].

In relation to the *TS(P),* we have defined a group of characteristics related to the position into the environment, the possession (or not) of an object, as well as characteristics that identify the physical state (*PS*) or the moods (*M*) —using as reference the Plutchik Model[10]— that presents PANCHO in any moment. The *Transitory States* for an agent PANCHO will be represented as follows:

$$TS(P) = Pos(P) \cup Own\_Object(P) \cup PS(P) \cup M(P) \tag{3}$$

Finally, the attitudes selected to generate the team roles behaviors in an agent PANCHO will be defined as follows:

$$A(P) = \bigcup_{i=1}^{n} Trust(Ti) \cup \bigcup_{i=1}^{n} Esteem(Ti) \tag{4}$$

With n= number of trainees

On the other hand, the beliefs about the teammates for an agent PANCHO will be represented as beliefs about each one of the team members, and beliefs about the team as a whole.

$$Bel(HGroup) = \bigcup_{i=1}^{n} Bel(Ti) \cup Bel(HTeam) \tag{5}$$

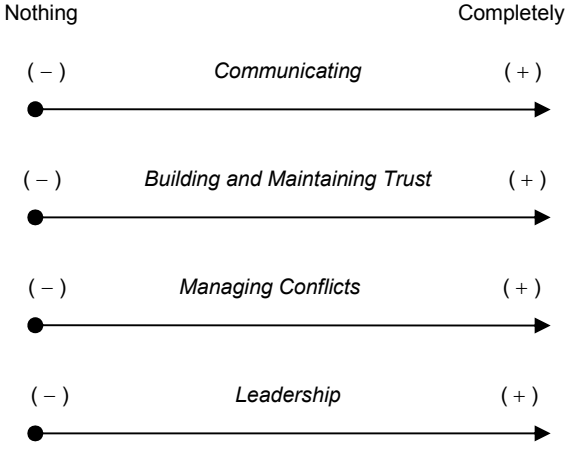
With n= number of trainees

The beliefs considered for each one of the teammates follow a similar organization to beliefs about himself, and will be defined as:

$$Bel(Ti) = DC(Ti) \cup TS(Ti) \tag{6}$$

Because the second goal in PANCHO is providing scaffolding to the trainees during the execution stage, we have included into the *DC(Ti)* beliefs about the *Collaborative Skills (CS)* [11] that the trainees own. These beliefs have been modeled in a previous stage of the TTS (*Planning Stage*) according to the mechanism described in [12].

The value for each one of these *CSs* corresponds with the degree of skill that the trainee has, and it will be represented as follows:



6 Conclusions and Ongoing Work

We have described an automated mechanism to enhance human teams replacing a human member with an IVA. This mechanism is part of a Team Training Stategy that we have developed. The IVA called PANCHO joins the human group with two goals: improving the performance of the group, incorporating a team role compatible with the other team members —a best Team Role for team composition— and providing scaffolding to the trainees during the task execution.

In order to do an experimental study for the evaluation of the *Execution stage*, according to TTS, we have implemented a CVE with ludic characteristics. PANCHO has been instantiated in this context —using the process proposed for *Training Domain Specification*— to perform two team roles: Coordinator and Implementer.

Acknowledgements

We thank the support of the Ministry of Education and Science (Spain) to the project ENVIRA (REF/TIN2006-15202-C03-01), as well as to the Secretariat of Public Education of the Yucatan State (Mexico) for the financial support in 2008.

References

1. Rickel, J., Johnson, W.L.: Virtual Humans for team Training in Virtual Reality. In: Proceedings of the 9th World Conference on AIED, pp. 578–585 (1999)
2. Aguilar, R.A., Troncoso, B., de Antonio, A., Imbert, R.: Intelligent Virtual Environments for Training: A Tutoring Approach. Research in Computing Science 27, 169–179 (2007)
3. Belbin, M.: Management Teams. John Wiley & Sons, New York (1981)
4. Belbin, M.: Team Roles at Work. Butterworth-Heinemann Ltd., London (1993)
5. Bean, J.C.: Engaging ideas: the profesor's guide to integrating writing, critical thinking, and active learning in the classroom. Jossey-Bass, San Francisco (1996)
6. McGath, J.E.: Groups: Interaction and Performance. Prentice Hall, New York (1984)
7. Imbert, R., de Antonio, A.: When Emotion Does Not Mean Loss of Control. In: Panayiotopoulos, T., et al. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 152–165. Springer, Heidelberg (2005)
8. Costa Jr., P.T., McCrae, R.R.: The NEO Personality inventory manual. Psychological Assessment Resources, Odessa (1985)
9. Lindgren, R.: R. Meredith Belbin's team roles viewed from the perspective of the Big 5: A content validation. Universidad de Oslo, Oslo (1997)
10. Plutchik, R.: A general psychoevolutionary theory of emotion. In: Plutchik, R., Kellerman, H. (eds.) Emotion: Theory, research, and experience. Theories of emotion, vol. 1, pp. 3–33. Academic, New York (1980)
11. Johnson, D., Johnson, R.: Learning Together and Alone. Prentice Hall, Englewood Cliffs (1994)
12. Aguilar, R.A., de Antonio, A., Imbert, R.: PANCHO needs Models of Collaborative Human Groups: A Mechanism for Teams Modeling. Research in Computing Science 34, 299–310 (2008)

Authoring Neuro-fuzzy Tutoring Systems for M and E-Learning

Ramón Zatarain-Cabada[1], M.L. Barrón-Estrada[1], Guillermo Sandoval[1], Moisés Osorio[1], Eduardo Urías[1], and Carlos A. Reyes-García[2]

[1] Instituto Tecnológico de Culiacán
Juan de Dios Bátiz s/n, col. Guadalupe, Culiacán, Sinaloa, 80220, México
rzatarain@itculiacan.edu.mx
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)
Luis Enrique Erro No. 1, Sta. Ma. Tonanzintla, Puebla, C.P. 72840, México
kargaxxi@inaoep.mx

Abstract. This paper is about an author tool that can be used to produce neuro-fuzzy tutoring systems for distance and mobile environments. These tutoring systems recognize and classify learning characteristics of learners by using a neuro-fuzzy system. The author tool has three main components: a content editor for building course structure and learning material; an editor for building fuzzy sets for different linguistic variables; and an XML course interpreter which combines a neuro-fuzzy predictive algorithm to display contents on different learning platforms. The author tool builds learning objects from other learning objects which are exported to SCORM format or to mobile devices.

Keywords: M-Learning, Intelligent Tutoring System, Distance Learning, Author Tool.

1 Introduction

Intelligent Tutoring Systems (ITS) build deductions about a learner's mastery of topics in order to dynamically adapt the content or style of instruction. They allow students having more control over their learning. In the last years they have proven to be highly successful [1]. However, they are still difficult to design and to implement. Authoring Intelligent Tutoring Systems are one of the answers for that difficulty. Today, many different authoring systems have been built. They can be classified [2] according to the type of ITSs they produce (Tutoring strategies, device simulation, domain expert systems, etc.).

On the other hand, with the propagation of mobile technology in society and industry, many organizations are using mobile devices for education [3, 4, 5]. New tools can be used by course authors not only to define the contents of a course, but also to create adaptive or personalized learning systems that are able of supervising and controlling the entire learning process [6, 7]. When these systems incorporate wireless features, the term m-learning is employed. Nonetheless, the main benefit of m-learning is to provide learning resources to learners "anywhere and anytime".

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 789–796, 2008.

In this paper, we present **HLTutor**, an author tool which can be used to build personalized or intelligent tutoring courses to be used in both learning settings: distance and mobile learning. The tool can be employed for developing learning material using SCORM [8] Learning Objects or other standard file formats. The output of the author tool will be either learning material for mobile devices or SCORM learning objects for e-learning environments. The learning material for mobile tools uses a neuro-fuzzy mechanism to identify and to predict learning styles in order to provide an adapted learning scheme. The learning styles are based on Gardner's Pedagogical Model of Multiple Intelligences [9].

The arrangement of the paper is as follows: In Section 2, we describe the general structure of the tool. In Section 3, we describe the neuro-fuzzy predictive model. Tests and results are shown in Section 4. Comparison to related work is given in section 5 and conclusions and future work are discussed in Section 6.

2 Hltutor General Structure

Figure 1 presents the general structure of **HLTutor**. As shown in Figure 1, the tool has two main editors: the **content editor** and the **fuzzy set editor**. An author creates a tutoring system by first building a course structure using the content editor. This structure consists of a number of units where a unit can be linked with learning material and some assessments. A course is created by importing already prepared learning material in different standard formats like **html, pdf, doc** or **SCORM** [8] learning objects from any type of source. The author can also introduce learning material by using a simple editor included in the tool. When the author introduces the learning material he/she creates four different instances corresponding to four different student learning styles (types) according to Gardner's Pedagogical Model of Multiple Intelligences: Logical/Mathematical, Verbal/Linguistic, Visual/Spatial and Musical/Rhythmic. There is a special interface in the content editor for helping the author when building this material.

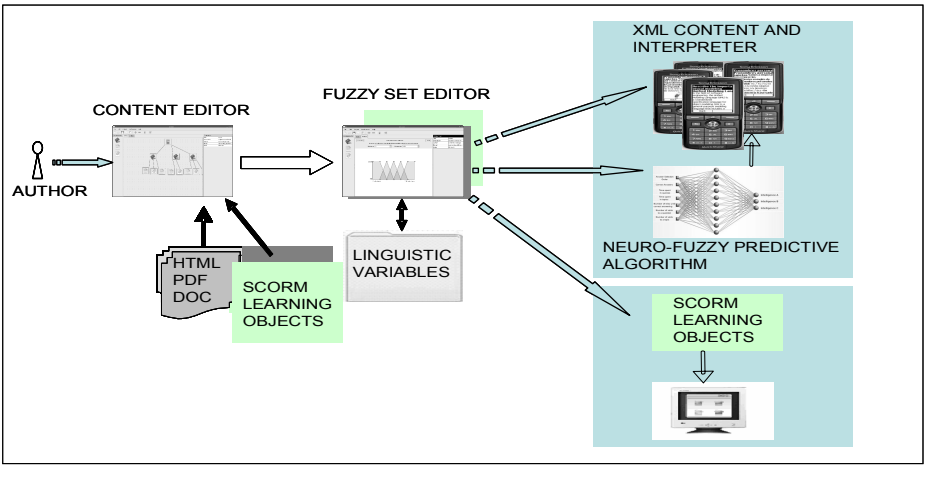


Fig. 1. General Structure of HLTutor

The fuzzy set editor helps the user to define **Fuzzy Membership Functions**. For **Fuzzy Inputs** there are seven linguistic variables defined for a user. They are: *answer selection order, correct answers, quiz spent time, topic spent time, number of tries until correct answer, number of visits to a question, and number of visits to a topic*. Each of the seven linguistic variables allows three different values: low, average, and high. A Fuzzy inference process is performed mapping input linguistic values to output multiple-intelligence styles. The output of the fuzzy set editor can be m-learning material in **XML format** along with a predictive engine that employs a **neuro-fuzzy inference algorithm** [10]. It operates online using present and former information for each individual learner. At the start of each learning unit, predictions are made as to what the learners preferred learning material is, based on former information built in the fuzzy set edition phase. Afterward, the neuro-fuzzy algorithm will learn from present information and will make adjustments, if necessary, to another best suited learning style. Another option to the output of the fuzzy editor is to export the learning material to SCORM format. The benefit of this format is the availability of the material in any distance learning environment.

When a mobile course is exported to a mobile device, a **XML interpreter** is added to the course. This interpreter has the job of displaying the material of the course into the mobile device, according to some chosen learning style.

3 Neuro-fuzzy Predictive Model

Figure 2 shows part of the HLTutor Neuro-Fuzzy system (just two linguistic variables are shown) implemented in order to represent knowledge (seven linguistic variables, four multiple intelligences, and inference rules), to learn from former and current data, and to make adjustment to new learning styles.

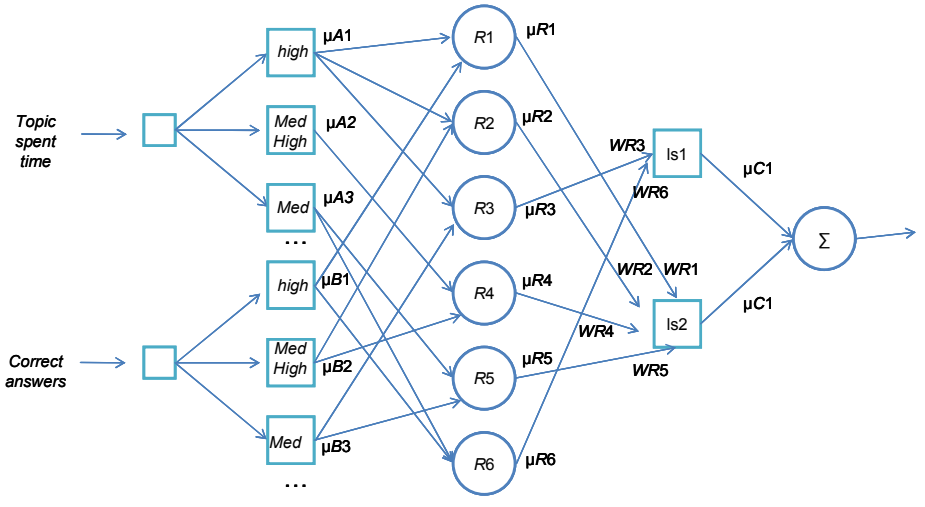


Fig. 2. Part of the HLTutor Neuro-Fuzzy System

As we can observe in Figure 2, the system configuration consists of input and output layers, and three hidden layers that represent membership functions and fuzzy rules [10]. The complete **input layer** has seven neurons representing our seven linguistic variables. Every linguistic variable has three fuzzy sets (low, average, and high). The input layer sends out external crisp values directly to the next layer. The output of this layer is as follows:

$$y_i = x_i ,$$

where x_i and y_i are input and output respectively.

Each neuron of the input layer is connected to three neurons on **layer 2** (fuzzyfication layer). In this layer crisp values from input layer are transformed to appropriate linguistic fuzzy sets (*low, average, or high sets*). Every neuron of layer 2 represents a fuzzy set for each one of the seven linguistic variables (see table 1*)*. The output of the layer is the degrees of membership of each input value to each fuzzy set. Every neuron of **Layer 3** or **fuzzy rule layer** represents a fuzzy rule (R1, R2, R3, etc.). Every fuzzy rule neuron takes inputs from layer 2 neurons. The rule is evaluated by the fuzzy **intersection** or **t-norm**, which in this case is the product operation. The output of each neuron is described as:

$$y_i = x_{1i} \times x_{2i} \times \ldots \times x_{ki}$$

The output of layer 3 represents the weights of each one of the rules. The weights connecting layer 3 and layer 4 are changed or adjusted by training the neural network. The weight values are normalized on many adjustments, by dividing each weight into the greatest weight found on each one of the adjustments or iterations and is represented by

$$w_{ni}(p + 1) = w_i(p) / w_{max}(p),$$

where w_{ni} is the normalized weight, p is the last iteration, w_i is the weight of connection i, and w_{max} is the greatest weight of the iteration.

Table 1. Fuzzy sets for two Linguistic Variables

TopicsSpentTime	CorrectAnswers
Range=[0 1000] NumMFs=5 MF1='Low':'trapmf',[-225 -25 100 300] MF2='Medium-Low':'trimf',[100 300 500] MF3='Medium':'trimf',[300 500 700] MF4='Medium-High':'trimf',[500 700 900] MF5='High':'trapmf',[700 900 1025 1225]	Range=[0 10] NumMFs=5 MF1='Low':'trapmf',[-2.25 -0.25 1 3] MF2='Medium-Low':'trimf',[1 3 5] MF3='Medium':'trimf',[3 5 7] MF4='Medium-High':'trimf',[5 7 9] MF5='High':'trapmf',[7 9 10.2 12.2]

Layer 4 or **output membership layer** takes inputs from the fuzzy rule neurons and merges them by using fuzzy **union**, which in our case is the algebraic sum and is defined as

$$x_{1i} \oplus x_{2i} \oplus \ldots \oplus x_{ji}$$

The output of layer 4 are the different learning styles (Logical/Mathematical, Verbal/Linguistic, Visual/Spatial and Musical/Rhythmic) produced in the fuzzy rule layer.

Layer 5 or **defuzzyfication layer** is the output of our system. In this layer the **sum-product composition** method is used [11]. The output of this last layer is a recommended learning style for the student. The learning algorithm we used is **back-propagation**. Our algorithm takes a desired output (a default learning style), which is computed at the beginning of the iterations. Next, the algorithm computes the actual output and compares it with the desired output (which is dynamically adjusted according with the student test results). If there is a difference between the actual and the desired output, the error is calculated and propagated backward through the network, updating or adjusting the weights between connected neurons. The neural network was trained too using Matlab (version 7.1). Then, the trained neural network was implemented using Java along with the XML interpreter.

4 Tests and Results

We conducted a test of our tool by working with a group of different kinds of users (authors), mainly university professors, and college students. They produced personalized or intelligent tutoring courses for English Language (Elementary school), Object-Oriented Analysis and Design, and a Basic Math Course. The courses dynamically adapt the content or style of instruction presented to the learner. Figure 3

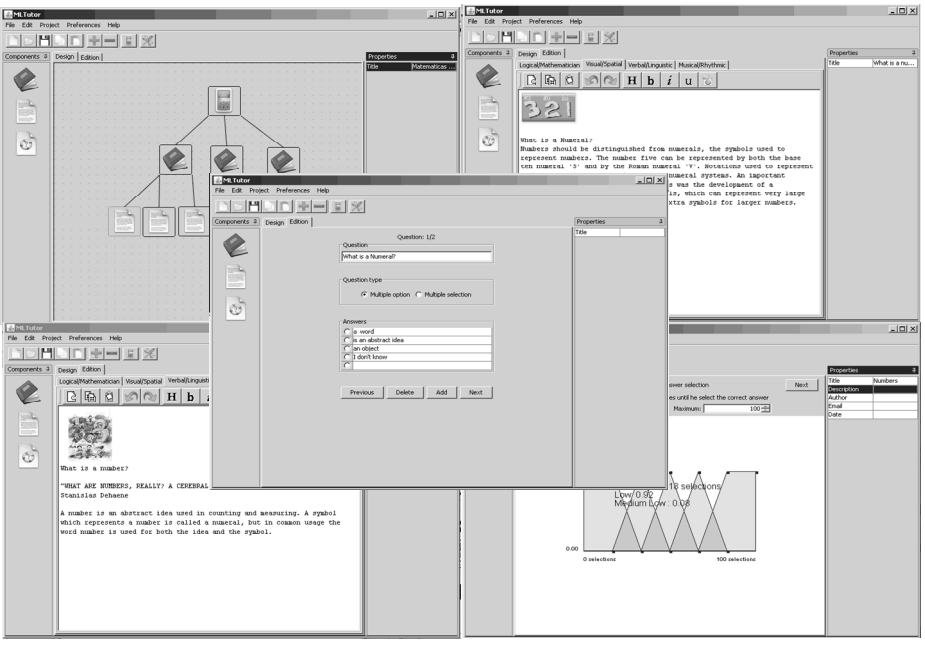


Fig. 3. HLTutor Interfaces

Fig. 4. Math Course in a Cell Phone

shows some interfaces (snapshots) of the author tool when building the Basic Math course. As we can observe, one image shows the course tree structure (left-top); two images show the contents of the course under the Verbal/Linguistic type of Intelligence (right-top and left-bottom); another one illustrates the fuzzy set edition (right-bottom); and the last one (front image) illustrates a quiz edition.

Figure 4 presents some instances of the Math course running in a cell Phone (NHAL Win32 Emulator). The four mobile instances correspond (left-to-right, top-bottom) to *Three-Chapter menu, Chapter-1 menu (contents and quiz), chapter-1 contents, and chapter-1 quiz*.

5 Related Work

None of the ITS authoring tools listed by [2] produce tutoring systems for mobile environments. Some of the systems produced by adaptive hypermedia authoring tools are more Web-oriented. There are several author tools used to create mobile applications like *MyLearning* [12], *zirada*[13], Test Editor [7], or mediaBoard[4]. Some of them are more PocketPC's oriented and some are more focused to quiz editing or game-based learning. But, none of those author tools have the capability of adaptation to the user's form of learning and the tool's portability across different computer and operating system platforms.

6 Conclusions and Future Work

We presented an author tool that allows an author to produce personalized learning material to be used under distance and mobile learning environments. The software tool was implemented with Java version 1.6 and XML. The learning material produced with the tool is platform independent and it has been tested with different software emulators (NHAL Win32 Emulator , J2ME Wireless Toolkit and Sony Ericsson SDK) and with different cell and smart phones (Sony Ericsson and Nokia). We presented a test with one of them. Currently empirical studies are taking place to examine the reaction of students to the learning material produced with HLTutor. The study continues examining different strategies like mixing material of one course under two learning environments (e-Learning and m-Learning). The purpose of these studies is to examine the reaction and learning performance with different learning environments. Future work should focus on more testing with official and complete material for course programs in public and private schools in Mexico.

Aknowledgments

The work described in this paper is fully supported by a grant from the DGEST (Dirección General de Educación Superior Tecnológica) in México under the program "support and development of academic bodies" [Academic Body: Research in Software Engineering].

References

[1] Koedinger, K., Anderson, J.: Intelligent tutoring goes to the big city. In: Greer, J. (ed.) Proceedings of the International Conference on Artificial Intelligence in Education, pp. 421–428. AACE, Charlottesville (1995)

[2] Murray, T.: Authoring Intelligent Tutoring Systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education 10, 98–129 (1999)

[3] Bull, S., Cui, Y., McEnvoy, A.T., Reid, E., Yang, W.: Roles for Mobile Learners Models. In: Proceedings of the 2nd IEEE Workshop on Wireless and Mobile Technologies in Education (WMTE 2004), pp. 124–128. IEEE Computer Society, Los Alamitos (2004)

[4] Attewell, J.: From Research and Development to Mobile Learning: Tools for Education and Training Providers and their Learners. In: Proceedings of the 4th World Conference on Mobile Learning (mLearn 2005) (2005)

[5] Anderson, P., Blackwood, A.: Mobile and PDA technologies and their future use in education, JISC Technology and Standards Watch Report (2004)

[6] Shih, K.-P., Chang, C.-Y., Chen, H.-C., Wang., S.-S.: A Self-Regulated Learning System with Scaffolding Support for Self-Regulated e/m-Learning. In: Proceedings of the 3rd International Conference on Information Technology: Research and Education (ITRE 2005), pp. 30–34. IEEE Computer Society, Los Alamitos (2005)

[7] Romero, C., Ventura, S., Hervás, C., De Bra, P.: An Authoring Tool for Building Both Mobile Adaptable Tests and Web-Based Adaptive or Classic Tests. In: Wade, V.P., Ashman, H., Smyth, B. (eds.) AH 2006. LNCS, vol. 4018, pp. 203–212. Springer, Heidelberg (2006)

[8] SCORM, The Sharable Content Object Reference Model, February 14 (2008), http://www.adlnet.org
[9] Gardner, H.: Frames of Mind: The theory of multiple intelligences. Basic Books, New York (1983)
[10] Negnevitsky, M.: Artificial Intelligence: A guide to Intelligent Systems, 2nd edn. Addison-Wesley, Reading (2005)
[11] Jang, J.: ANFIS: Adaptive Network-based Fuzzy Inference Systems. IEEE Transactions on Systems, Man and Cybernetics 23(3), 665–685 (1993)
[12] Attewell, J.: Mobile technologies and learning: A technology update and mlearning project summary, February 14, 2008. Learning and Skills Development Agency, London (2008),
[13] http://www.m-learning.org/reports.shtml
[14] Zirada Mobile Publisher, The Premier Mobile Content Creation Tool, February 14 (2008), http://www.Zirada.com

Combining Concept Maps and Petri Nets to Generate Intelligent Tutoring Systems: A Possible Approach

Maikel León, Isis Bonet, María M. García, and Zenaida García

Department of Computer Science, Central University of Las Villas (UCLV)
Highway to Camajuaní km 5 ½, Santa Clara, Cuba
{mle,isisb,mmgarcia,zgarcia}@uclv.edu.cu

Abstract. The use of pedagogical methods with the technologies of the information and communications produces a new quality that favours the task of generating, transmitting and sharing knowledge. Such is the case of the pedagogical effect that produces the use of the Concept Maps, which constitute a tool for the management of knowledge, an aid to personalize the learning process, to exchange knowledge, and to learn how to learn. In this paper the authors present a new approach to elaborate Intelligent Tutoring Systems, where the techniques of Concept Maps and Artificial Intelligence are combined, using Petri Nets as theoretical frame, for the student model. The pedagogical model that controls the interaction between the apprentice and the generated Intelligent Tutoring Systems is implemented by Petri Nets. The Petri Nets transitions are controlled by conditions that refer to the apprentice model. The firing of these transitions produces actions that update this apprentice model. These conditions are automatically included into the pedagogical model and the teacher has only to specify the contents of the domain model.

1 Introduction

To construct and to share knowledge, to learn significantly, and to learn how to learn, are ideas on which researchers have been pondering for a long time as well as the use of tools that would allow taking these aspirations into practice. To achieve this, different techniques and strategies have been used. Concept Maps (CM) provide a schematic summary of what is learned and they order it in a hierarchic range. The knowledge is organized and represented in all the levels of abstraction, the general knowledge goes to the upper part and the most specific one goes to the lower [1]. Over the last few years the CM have increasingly got a great popularity and its integration with the technologies of the information and the communications have become a very important element in the plans for the improvement of the teaching-learning process. The main application of the CM has had effect in the teaching-learning process, which is its basic intention, they are based on an instrument that combines the scientific rigidity with simplicity and flexibility, producing a general approval in the audience of students and professionals; this represents an important nexus between pedagogy and technology in a huge field of applications. Also, it constitutes an important aid for those who generate, transmit, store, and divulge information and knowledge and it comprises an

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 797–805, 2008.

important tool to obtain a highest practical value in the systems of the teaching-learning process [2].

Thus, Petri Nets (PN) are a graphical and mathematical modelling tool. It consists of places, transitions, and arcs that connect them. Input arcs connect places with transitions, while output arcs start at a transition and end at a place. There are other types of arcs, e.g. inhibitor arcs. Places can contain tokens; the current state of the modelled system (the marking) is given by the number (and type if the tokens are distinguishable) of tokens in each place. Transitions are active components. They model activities which can occur (the transition fires), thus changing the state of the system (the marking of the PN). Transitions are only allowed to fire if they are enabled, which means that all the preconditions for the activity must be fulfilled (there are enough tokens available in the input places). When the transition fires, it removes tokens from its input places and adds some at all of its output places. The number of tokens removed/added depends on the cardinality of each arc. The interactive firing of transitions in subsequent markings is called token game [3]. PN are a promising tool for describing and studying systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. As a graphical tool, PN can be used as a visual-communication aid similar to flow charts, block diagrams, and networks. In addition, tokens are used in these nets to simulate the dynamic and concurrent activities of systems. As a mathematical tool, it is possible to set up state equations, algebraic equations, and other mathematical models governing the behaviour of systems [4].

To study performance and dependability issues of systems it is necessary to include a timing concept into the model. There are several possibilities to do this for a PN; however, the most common way is to associate a firing delay with each transition. This delay specifies the time that the transition has to be enabled, before it can actually fire. If the delay is a random distribution function, the resulting net class is called stochastic PN. Different types of transitions can be distinguished depending on their associated delay, for instance immediate transitions (no delay), exponential transitions (delay is an exponential distribution), and deterministic transitions (delay is fixed). The concept of PN has its origin in Carl Adam Petri's dissertation Kommunikation mit Automaten, submitted in 1962 to the faculty of Mathematics and Physics at the Technische Universität Darmstadt, Germany.

Intelligent Tutoring Systems (ITS) have demonstrated its effectiveness in diverse domains. However, its construction implies a complex and intense work of engineering of knowledge, which prevents an effective use of it. In order to eliminate these deficiencies, appears the target to construct the author's hardware which facilitates the construction of these systems to all users (not necessarily expert) in the computer field, in particular to those instructors who master a certain matter of education [5]. The field of the ITS is characterized by the application of Artificial Intelligence techniques, to the development of the teaching-learning process assisted by computers. If the key feature of an ITS is the aptitude to adapt itself to the student, the key component of the system is the Student Model, where the information associated to the student is stored. This information must be inferred by the system depending on the information available: previous data, response to questions, etc. This process of inference is identified as diagnosis, uses the Artificial Intelligence techniques to represent knowledge, to shape the human reasoning, to emphasize the learning by means of the

action, to combine experiences of resolution and discovery, to be able to solve problems by their own, to formulate diagnoses and to provide explanations. So, they count on a bank of instruction strategies which helps to decide what and how to inform to the student to get an effective direction.

The aim of this paper is to present a new approach to elaborate ITS using a CM, a questionnaires able to catch the cognitive and affective state of the student (Student Model) and by using PN to allow the student to sail in an oriented way according to their knowledge and not in a free way as the CM are conceived. And it is the capacity to adapt dynamically to the development of the student's learning what makes this system intelligent. The pedagogical model is controlled by PN that access and updates the apprentice model, while presenting the contents of the domain model. The PN are automatically generated from specifications defined by the teacher using a graphical interface and compiled into a neural net that actually implement the ITS. An important point in the proposed approach is that the teacher does not need to specify the interaction with the apprentice model; this interaction is automatically included in the PN using the prerequisite and difficulty orders previously defined by the teacher.

2 Motivations for Concept Maps

Concept Maps provide a framework for capturing experts' internal knowledge and making it explicit in a visual, graphical form that can be easily examined and shared. CM constitute a tool of great profit for teachers, investigators of educational topics, psychologists, sociologists and students in general, as well as for other areas especially when it is necessary to manage with large volumes of information. They have become a very important element in the plans for the improvement of the ITS and they have also extended its use to other areas of the human activity in which both management and the use of knowledge take up a preponderant place. If to define a CM is relatively simple, it is simpler to understand the meaning of the definition. The Concept Maps were defined by Novak, his creator, as a skill that represents a strategy of learning, a method to get the gist of a topic and a schematic resource to represent a set of conceptual meanings included in a set of propositions [2].

It is necessary to point out that there is not only one model of CM, several may exist. The important point is the relations that are established between the concepts through the linking words to form propositions that configure a real value on the studied object. For such a reason, in a concept there may appear diversity of real values. In fact, it turns very difficult to find two exactly equal CM, due to the individual character of knowledge. The CM can be described under diverse perspectives: abstract, visualization, and conversation. Since significant learning is reached more easily when the new concepts or conceptual meanings are included under wider concepts, the most used CM are the hierarchic ones, the most general and inclusive concepts are placed in the upper part of the map, and the progressively more specific and less inclusive concepts, in the lower part. The subordinated relations among concepts may change in different fragments of learning, so in a CM, any concept may rise up to the top position, and keep a significant propositional relation with other concepts of the map. The use of CM designed by the professor increases both learning and retention of scientific information. The students produce maps as learning tools. Considering

that the CM constitute an explicit and clear representation of the concepts and propositions, they allow both teachers and students to exchange points of view on the validity of a specific propositional link and to recognize the missing connections in the concepts that suggest the need of a new learning. For this reason, this skill has complemented so favourably with the practice of distance learning which presupposes that students and teachers are not physically in the same place at the same time.

Concept Maps have particular characteristics that make them amenable to smart tools. These include [2]:

1. Concept Maps have structure: By definition, more general concepts are presented at the top with more specific concepts at the bottom. Other structural information, e.g. the number of ingoing and outgoing links of a concept, may provide additional information regarding a concept's role in the map.
2. Concept Maps are based on propositions: every two concepts with their linking phrase forms a "unit of meaning". This propositional structure distinguishes Concept Maps from other tools such as Mind Mapping and The Brain, and provides semantics to the relationships between concepts.
3. Concept Maps have a context: A Concept Map is a representation of a person' understanding of a particular domain of knowledge. As such, all concepts and linking phrases are to be interpreted within that context.
4. Concepts and linking phrases are as short as possible, possibly single words.
5. Every two concepts joined by a linking phrase form a standalone proposition. The proposition can be read independently of the map and still "make sense".
6. The structure is hierarchical and the root node of the map is a good representative of the topic of the map.

The students who analyze CM will have a wider basic knowledge; therefore, they will be more prepared to solve problems in comparison to those students who learn by memorizing. The CM have turned into a useful instrument for teacher training and for the student's understanding of diverse subjects. CM constitute an instrument that merges the scientific rigidity with the simplicity and flexibility. It represents an aid for those who generate, transmit, store and spread information and knowledge. They also constitute an important tool to achieve a practical value, especially in ITS.

3 Proposed Model

The ITS that are created correspond with a CM, with the particular feature that in some of its nodes there appears a questionnaire, capable to get the cognitive and affective state of the student and able to guide his navigation, creating this way an "Intelligent" CM. At the first level of the authoring interface, the teacher must specify the set of pedagogical units that define each curriculum and, at the second level, the set of problems that define each pedagogical unit. The elements of these sets present a partial order and associated prerequisites [6]. To specify these partial orders and prerequisites the teacher disposes of a graphical interface that allows the construction of graphs. In these graphs, an edge from node n_1 to node n_2 means that n_2 has n_1 as a prerequisite. Each node n may have the following input edges:

- None: node n has no prerequisite and can be executed anytime. This is the case only for initial nodes.
- One: node n has only one node as prerequisite and this one must be executed before node n is available for execution.
- Two or more necessary edges: node n has several prerequisite nodes and all of them must be executed, before node n is available for execution.
- Two or more alternative edges: node n has several prerequisite nodes but only one of them must be executed before node n is available for execution.

Necessary and alternative edges may occur simultaneously in the same node. Nodes and the different types of edges may be combined in a complex graph, according to the intended course sequences, but they must satisfy one restriction: each graph must have only one initial node and only one final node. This restriction seems reasonable, because pedagogical units and problems should have one begin and one end point and it also assures that the graphs of the two levels can be combined into a one level PN. Figure 1 shows a prerequisite graph with "AND" and "OR" constraints associated with necessary and alternative edges.

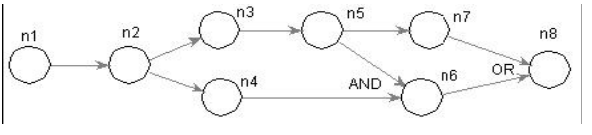


Fig. 1. Prerequisites Graph

The first step in the generation of the final code of an ITS using the proposed authoring tool is to compile each prerequisite graph into a PN that represents the pedagogical model. The pedagogical model defines the control strategy of the interaction between the apprentice. To represent the pedagogical model is used an Object Petri Net (OPN) [4]. An OPN is defined by a control structure (places, transitions and arcs connecting places to/from transitions), by the data structure of its tokens and by its semantics, that defines how the token player should execute the OPN, given an initial marking. The OPN places can represent either pedagogical unit, if the graph corresponds to the first level of the interface, or problems, if it corresponds to the second level. The net structure is directly obtained from the graphs specified by the teacher and the token data structure is defined by the adopted apprentice model. As the tokens have the class Apprentice the marking of the OPN is associated with the current activity of the apprentice and also with the possible future activities. A transition fires when the apprentice has finished the interactions associated with its input nodes. The token is then moved to the output places of the transition, meaning that the apprentice is ready to begin the activities associated with them. A prerequisite graph (curriculum or pedagogical unit) is automatically translated into an OPN structure. Each configuration of nodes and edges in a prerequisite graph is associated with an OPN fragment that implements it and these fragments are combined into a single Petri net that implements the graph.

For instance, a simple sequence of nodes, see figure 2(a), is implemented by one transition (t1 2), one input place (n1) and one output place (n2). One node with two

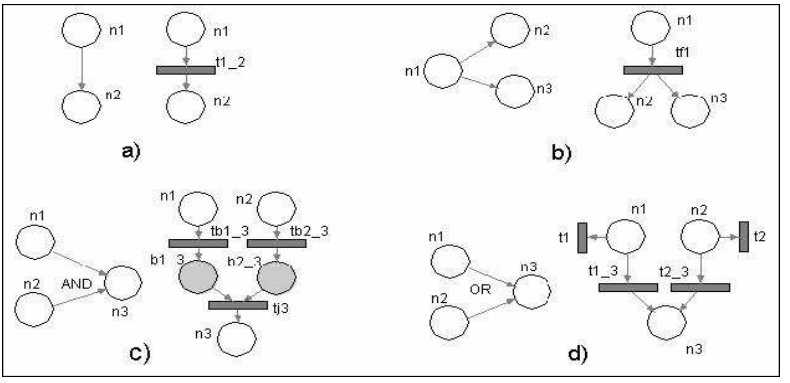


Fig. 2. Graph Topologies and Petri Nets

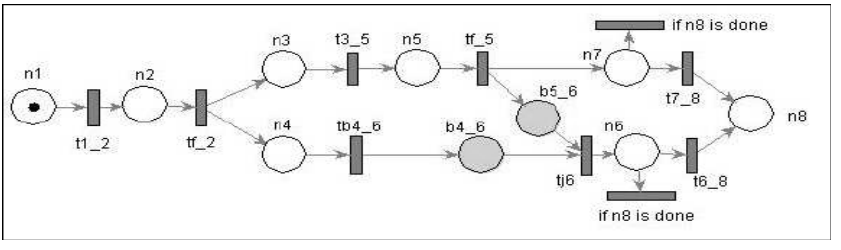


Fig. 3. Structure of an OPN

output edges, see figure 2(b), is implemented by a fork transition (tf1) with one input place (n1) and two output places (n2 and n3). After the activity associated with the input place is finished, the apprentice can do either the activity associated with one output place or the activity associated with the other. A node with two input necessary edges, see figure 2(c), is implemented by a join transition (tj3) with two input places (b1 3 and b2 3) and one output place (n3). The places b1 3 and b2 3 are buffer places and do not correspond to pedagogical units or problems. They are necessary because after the activity associated with either one of the places n1 or n2 is finished, that information must be stored in the corresponding buffer and only when the activity associated with the other place is also finished (and also stored in the corresponding buffer) the activity associated with the output place (n3) can be performed. For this, we need two extra transitions (tb1 3 and tb2 3) connecting, respectively, the input places n1 and n2 to the buffers b1 3 and b2 3. Finally, see figure 2(d), a node with two alternative input edges is represented by two transitions (t1 3 and t2 3), with input places n1 and n2, respectively, and both with output place n3. Transitions t1 and t2 are used to remove the apprentice token from places n1 and n2, respectively; in the case the activities associated with n3 have already been performed, avoiding in this way the repetition of n3. Figure 3 shows the OPN obtained from the prerequisite graph of figure 1.

The semantic of an OPN is the particular way the token player executes this net given an initial marking. An ordinary token player (used for PN with binary tokens)

verifies, from an initial marking, whether there are enabled transitions. If there is just one, the token player fires it. If there is more than one, it chooses one of them according to a priority order, or arbitrarily if no priority order is specified. In an OPN token player, because the token is associated with a data structure, the transitions can have pre-conditions that refer to the token attributes and/or to external variables [4]. The apprentice token attributes that are relevant to the definition of algorithm are name, doing and tobedone and the relevant external variables are next, that contains the next activity to be performed, and answer, that receives the interaction support unit returned values. We say that the apprentice, identified by name, is doing a problem if the apprentice is interacting with one of its associated interaction support units. When the apprentice finishes the interaction with one support unit, the unit returns one of the following navigation commands: halt, continue, repeat, that is stored in the external variable answer.

The command halt means that the apprentice wants to interrupt the present session with the ITS, the command repeat means that the next activity should be another interaction support unit of the same problem and the command continue means that the current problem is finished and a next one must be chosen, by the coordinator or by the apprentice, who is asked to choose the next problem, if there is more than one problem in tobedone. The apprentice must choose only one problem to do next, only one transition can fire at a given moment [7]. The coordinator initializes the associated PN with a marking where the apprentice token is stored in the place associated with the first problem of the first pedagogical unit and calls the OPN token player.

4 Petri Nets and Neural Networks: A Possible Solution

PN are powerful and versatile tools for modelling, simulating, analyzing and designing of complex workflow systems. This topic mainly discusses a hybrid approach using neural network and PN in the formal model of an ITS. Neural networks are used in different fields; classification is one of problems where they are commonly used [8]. PN are alternative tools for the study of non-deterministic, concurrent, parallel, asynchronous, distributed or stochastic systems [9]. They can model systems in an easy and natural way. PN approach can be easily combined with other techniques and theories such as fuzzy theory, neural networks, etc [10]. Since PN offer advantages to model systems and can interact with other techniques easily, it would be advantageous to model neural networks starting from PN models, which allow not only the design adjustment but also the initialization of the neural network weights [11]. Following the algorithm in [4], we can model a neural network starting from a PN with the application of weighty production rules in the algorithm. See figure 4 and 5.

The following weighting production rules are obtained starting from the example of Figure 4 and following the algorithm described in [4]:

1. If P1 and P4 then P5 (w1, w4)
2. If P1 and P4 then P6 (w1, w4)
3. If P1 and P4 then P12 (w1, w4)
4. If P5 or P6 then P9 (μ4, μ5-9)
5. If P6 and P1 then P15 (w6, w1)
6. If P9 then P10 (μ6)

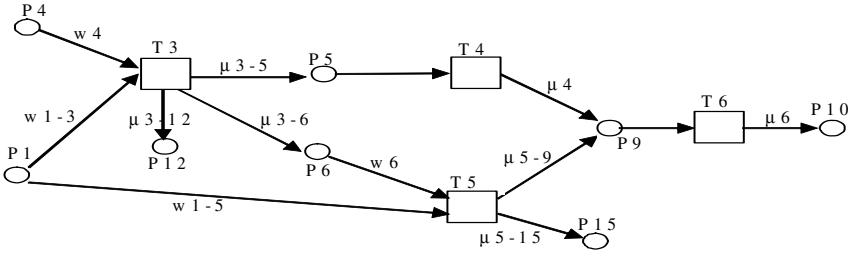


Fig. 4. Example of Petri Net

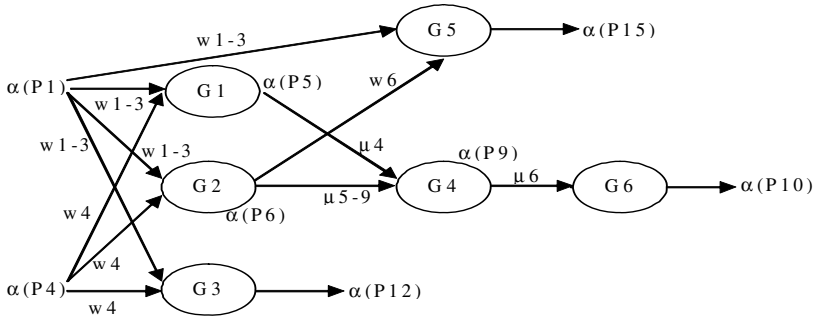


Fig. 5. A neural network model obtained from the Petri Net of Figure 4

The learning algorithm of the neural networks obtained is the same as the backpropagation of multilayer neural networks. The main idea is that all layer weights can be updated through the backpropagation algorithm if certainty factors of all sink places are given. A complex neural network can be divided into several sub-networks starting from the modular design of an original PN. A teacher can edit an ITS starting from a CM that will be compiled into a PN, the neural network that control the system interaction is constructed starting from the PN. The initial modelling PN will allow us to design a feedforward neural network with the backpropagation learning algorithm [12].

5 Conclusions

We propose an approach to model and to develop ITS that could be considered in the construction of the Student Model, where are combined the facilities of the CM for the organization of the knowledge and the potentiality of the PN that allows to the student to sail in an oriented way according to their knowledge and not in a free way as the CM are conceived. The suggested hybrid system combines PN and neural networks using the advantages of PN in order to overcome the neural network deficiencies concerning their original design and definition of their initial weighs.

References

1. Cañas Alberto, J., Carvalho, M.: Concept Maps and AI: an Unlikely Marriage? In: SBIE 2004: Simpósio Brasileiro de Informática na Educação Manaus, Brasil (2004)
2. Martínez, N.: Mapas Conceptuales y Redes Bayesianas en los Sistemas de En-señanza Inteligentes. Congreso Internacional de Informática Educativa, Costa Rica (2006)
3. Liu, X.-Q., Wu, M., Chen, J.-X.: Knowledge aggregation and naviga-tion highlevel petri nets-based in e-learning. In: Proceedings International Conferenceon Machine Learning and Cybernetics, vol. 1, pp. 420–425 (2002)
4. Li, X., Yu, W.: Dynamic Knowledge Inference and Learning under Adaptive Fuzzy Petri Net Framework. IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and reviews 30(4) (2000)
5. Esposito, F., Ferilli, S., Basile, A., Di Mauro, N.: Inference of abduction theories for handling incompleteness in first-order learning. Knowledge and Information Systems (KAIS) Journal, 1–17 (2006), ISSN: 0219-1377
6. Brusilovsky, P.: From intelligent tutoring systems to web-based education. In: 5th International Conference on ITS, Canada. LNCS. Springer, Heidelberg (2000)
7. Murray, T.: Authoring intelligent tutoring systems: An analysis of the state of the art. Journal of Artificial Intelligence and Education 10 (1999)
8. Hilera, J.: Redes neuronales artificiales: Fundamentos, modelos y aplicaciones, Ed. Addison-Wesley Iberoamericana (1995)
9. Yu, X., Chen, G.: Dynamic Learning Rate Optimization of the Backpropagation Algorithm. IEEE Transaction on Neural Networks 6(3), 669–677 (1995)
10. Bello, R.: Aplicaciones de la Inteligencia Artificial, Ediciones de la Noche, Guadalajara, Jalisco, México (2002), ISBN: 970-27-0177-5
11. Liu, W., Hong, J.: Re-investigating Dempster's Idea on Evidence Combination. Knowledge and Information Systems: An International Journal 2(2), 223–241 (2000)
12. Cooley, J.: Data Preparation for mining World Wide Web browsing patterns. Knowledge and Information Systems (KAIS) Journal, 1–27 (1999), ISSN: 0219-1377

Cooperation Strategies for Pursuit Games: From a Greedy to an Evolutive Approach

Juan Reverte, Francisco Gallego, Rosana Satorre, and Faraón Llorens

Department of Computer Science and Artificial Intelligence
University of Alicante
{jreverte,fgallego,rosana,faraon}@dccia.ua.es

Abstract. Developing coodination among groups of agents is a big challenge in multi-agent systems. An appropriate enviroment to test new solutions is the prey-predator pursuit problem. As it is stated many times in literature, algorithms and conclusions obtained in this environment can be extended and applied to many particular problems. The first solutions for this problem proposed greedy algorithms that seemed to do the job. However, when concurrency is added to the environment it is clear that inter-agent communication and coordination is essential to achieve good results.

This paper proposes two new ways to achieve agent coodination. It starts extending a well-known greedy strategy to get the best of a greedy approach. Next, a simple coodination protocol for prey-sight notice is developed. Finally, under the need of better coordination, a Neuroevolution approach is used to improve the solution. With these solutions developed, experiments are carried out and performance measures are compared. Results show that each new step represents an improvement with respect to the previous one. In conclusion, we consider this approach to be a very promising one, with still room for discussion and more improvements.

Keywords: Multi-agent systems, Communication, Coordination, Neuroevolution.

1 Introduction

The Predator-prey problem (or pursuit domain) is an appropriate testbed for multi-agents systems [1]. It consists of world where a group of agents (called predators) aims to chase and surround another agent (called prey) that tries to evade them [2]. The goal of predator agents is to surround (capture) prey without touching it (i.e. occupying the adjacent cells), while the goal of the prey, as expected, is not to be captured.

This problem has been adressed many times in literature. Initially, Korf [3] proposed a greedy algorithm without inter-agent communication. His approach was to use a fitness function that combined 2 forces: each predator was "attracted" by the prey and "repelled" from the closest other predator. This solution kept predators away from other predators while they got closer to the prey.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 806–815, 2008.

The idea was to chase the prey arranging predators in an stretching circle. Korf concluded that the pursuit domain was easily solved with local greedy heuristics.

A great number of alternatives have emerged since Korf's. Haynes [4] used genetic programming to evolve coordinated predators. Haynes compared differences between communicating and non-communicating predators with respect to their success in capturing the prey. He also co-evolved predators and the prey and found that a prey following a straight, diagonal line in an infinite world was never captured unless it was slower than its pursuers. This demonstrated that for certain instantiations of the domain, Korf's heuristic was not enough. Tan [5] used Reinforcement Learning to improve cooperation in three ways: (1) sharing instantaneous information (sensation, action, rewards obtained), (2) sharing episodes of instantaneous information, and (3) sharing learnt policies. Tan showed that agents learn faster when they learn cooperatively than when they learn individually. Later, Jim and Giles [6] proposed a genetic algorithm and multi-agent communication through a blackboard. A really interesting alternative was proposed by Katayama et al. [7]. They integrated Analytic Hierarchy Process (AHP) into a Profit-Sharing algorithm. They gave primary knowledge to agents when they started their learning processes and proposed a way to progressively stop providing "hints" to agents as they grow.

Despite the great number of proposed solutions, there is still room for improvements in different instantiations of the pursuit domain. As Tan stated in his work [5], coordination algorithms or protocols tested under the pursuit domain may easily be ported to other autonomous agents domains in general. This paper presents a new proposal for improving cooperation between predators in the pursuit domain. The idea presented here is to mix the efficiency of greedy approaches with two coordination proposals: a simple sight notice protocol and an evolutionary coordination system based on Neuroevolution [8]. Results show that this is a promising approach that develops a very efficient coordination mechanism, with still room for more improvements.

This paper is organized as follows. Section 2 explains the characteristics of the pursuit domain. Section 3 presents the extension to Korf's greedy approach and the two new inter-agent coodination approaches. In section 4, results of the experiments carried out are analyzed. Finally, conclusions and future lines to follow are given in section 5.

2 The Pursuit Domain

Stone and Veloso [1] considered the pursuit domain as an interesting start point because it is easy to understand and difficult to master. Moreover, it is still popular because it is possible to create many different instances with different sorts of handicaps. The most classical environment consisted of a finite, discrete, grid world where 4 predators tried to capture 1 prey. In this environment, agents moved sequentially and two agents were not allowed to be on the same cell.

In order to realize our experiments we used Kok and Vlassis' Pursuit Domain Package (PDP)[9]. PDP is a software package that simulates a pursuit domain

environment and allows to modify its parameters to instantiate different experimental scenarios. Concretelly, PDP was tuned for the purposes of our research to show off these characteristics: (1) toroidal world with a discrete, orthogonal grid of squared cells, (2) availability for the agents to move to every adjacent cell (9 possible options), (3) concurrency in the execution and movement of predators, (4) limited field of vision (FOV) for agents, (5) agent's capability to communicate with others inside FOV, (6) capture method: 4 predators occupying the 4 orthogonally adjacent cells (i.e. north, south, east and west).

Defined this way, PDP has some challenges to face. The three most remarkables ones are: (1) Concurrency lets predators move to the same cell in the same timestep (i.e. they collide). If this occurs, colliding predators are penalized by replacing them randomly. (2) FOV makes exploration necessary, and (3) the toroidal world removes the possibility of cornering the prey.

3 Methodology

Initially, Korf [3] considered a solution quite simple yet effective. The approach was to consider an "attractive" force that pushed predators towards the prey. The method was to calculate this force as fitness function for each of the possible cells to go next, and finally select the more attractive one. This solution had the problem that predators piled up and disturbed themselves; then, it turned difficult to achieve the final surrounding capture position. Korf overcomed this problem considering a "repulsive" force which pushed each predator away from the nearest other predator. With this new force, predators attacked the prey more jointly, not piling themselves up.

The reduced number of cycles that predators took to capture the prey with Korf's method seemed good enough not to consider the necessity of improving it. However, the differences between the environment used by Korf and new environments like PDP [9] lead to reconsider it. For instance, Korf reported that his algorithm captured the prey in 119 cycles in average. The experiments we have run in the most similar conditions possible to Korf's inside PDP took 366 cycles in average. In this case, which is the best one for Korf, the toroidal world and the collisions between agents multiply time to capture the prey by 3. When conditions get worse, namely when the FOV of predators is reduced, the performance of Korf's approach deteriorates exponentially.

This means that it is necessary to extend Korf's algorithm to deal with the new issues of the environment. One possible way to extend it is to reconsider the way Korf treated attractive and repulsive forces between agents. In his proposal, predators where attracted by the prey and repelled by the nearest other predator. This leads to situations where one predator may be repelled directly against other predator, resulting in a collision. Then, the first approach to take is to make predators repel from all other predators. Equation (1) shows the fitness function used to do this. This function depends on the (x, y) coordinates of the cell and calculates distances from that cell to prey location (X_p, Y_p) and to other n predators locations (X_i, Y_i) using Manhattan Distance $d(x, y, x', y')$. To

balance the relative amount of repulsive forces against the attractive one, a scale constant $k \in [0, 1]$ is added. We will call Extended Korf (ExtKorf, for short) to the algorithm which works like Korf's but using the fitness function from Eq.(1).

$$f(x, y) = d(x, y, X_p, Y_p) - k \sum_{i=1}^{n} d(x, y, X_i, Y_i) \quad (1)$$

The Extended Korf algorithm dramatically outperforms results of the Korf algorithm. The main reason for this is that it reduces collisions between predators by an order of magnitude, thus avoiding penalties. Results supporting this are explained in Sect. 4.

3.1 Cascading Sight Notice (CSN)

In the environment where Korf did his experiments, communication between agents were not necessary, as he demonstrated. The main reason was that his agents were able to see the whole world at once. But, the more we limit the FOV of the predators the more they need to get more information to efficiently locate the prey. When predators have a reduced FOV, most of the time it happens that when some predators have already found the prey, others are still wandering around. This delay in founding the prey could be avoided if the predators were able to effectively tell where was the prey to others when they had found it.

Consider that an agent is located at (x, y) and having a FOV of f cells means that the agent is only able to perceive what happens in the set of cells $C = \{(x', y') \mid x - f \leq x' \leq x + f, y - f \leq y' \leq y + f\}$. Take into account that this refers to sensing in general, and not seeing in particular. Therefore, an agent is only able to communicate with other agents being inside its FOV. Moreover, agents never know their global location, nor global coordinates of other agents. They are only aware of the relative location of other agents with respect to them.

In strict sense, the probability of a predator indefinitely not finding the prey in these conditions is not 0, and that is definitely a problem to overcome. But communication is not as simple as telling others directly where is the prey; there is no way to do that. In order to communicate where the prey is, we propose a simple protocol called Cascading Sight Notice (CSN). The idea is that a predator P^0 seeing the prey Y has to communicate the relative location of Y that P^0 is perceiving to each other predator P^i that P^0 can see (i.e. P^i is inside the FOV of P^0). Then, each P^i not seeing Y could locate it by listening to P^0. P^i will then be aware of the relative location of P^0 with respect to P^i and also aware of the relative location of Y with respect to P^0. So, P^i will be able to calculate the relative location of Y with respect to P^i by adding the vectors of the two relative locations it will know. Then, when P^i will have located the prey, P^i will resend this new relative location to other predators in its FOV. The cycle will go on until no predator will be hearing or hearing predators will have already known where prey is (see Fig. 1).

This simple protocol lets predators with reduced FOV find the prey earlier than predators without communication do, and this turns into an improvement in

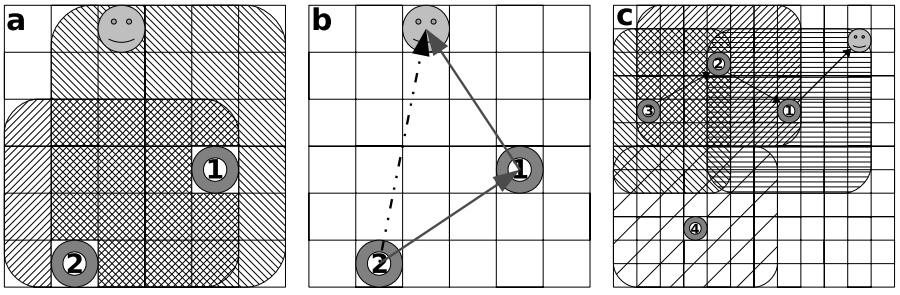


Fig. 1. a) Two predators with FOV 3 seeing each other, predator 1 seeing the prey. b) Predator 2 can figure out prey location from message of predator 1. c) Predators 1, 2 and 3 can figure out prey location, predator 4 cannot.

the average number of cycles needed to capture the prey. The results supporting this are shown and explained in Sect. 4 (see Fig.3).

3.2 NEAT Coordination Protocol (NECool)

Inside PDP, collisions occur when two or more predators move to the same cell on the same timestep. As long as predators decide where to move in a concurrent fashion, they have no opportunity of avoiding collisions unless they establish an appropriate coordination protocol. To create a fast and nearly-optimal coordination protocol we propose to evolve a neural network that decides the next movement to do for each predator. The neural network of predator P^i will receive as input the location of each predator P^j inside the FOV of P^i (including P^i), and a prediction of the next possible movements of each P^j. With this information, the neural network will be able to figure out the next movement that P^i should do to optimize the capture of the prey not colliding with any P^j.

The most relevant information each neural network will receive will be that regarding movement predictions of other predators. Each predator P^i has an associated 3×3 fitness matrix $F^i_{3\times3}$ where each element $F^i_{jk} \in [0, 1]$ represents the fitness associated to P^i moving to that cell, assuming the predator is currently located in F^i_{22}. The values of F^i are calculated using Extended Korf's algorithm, having previously used CSN to locate the prey.

These movement predictions represent 9 values for each predator, 36 for the 4 predators. Trying to pass these values directly to a neural network would require 36 input neurons, which means a great search space. Moreover, this approach is not scalable, because increasing the number of predators requires a linear increase in the number of input neurons. As long as a predator P^i is not interested in the exact F^j_{kl} of other predators P^j, but in the total other relation $F^j_{kl} < F^j_{mn}$, an approach to reduce dimensionality can be used (see Fig. 2). A

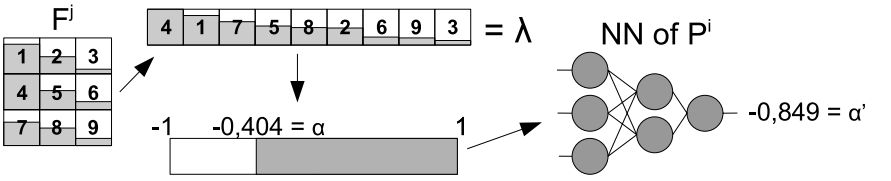


Fig. 2. Using Compressed Fitness Values to feed the neural networks

consecutive number from 1 to 9 is assigned to each cell of F^j. Then a vector with the 9 cells ordered by F^j_{kl} is formed. From the numbers assigned to the ordered cells in the vector, a 9-digit number $\lambda \in [123456789, 987654321]$ is formed. Then α is calculated as λ scaled into $[-1, 1]$. α is known as the Compressed Fitness Value, and it is sent to other predators as movement prediction. Then, each P^i feeds its neural network with the αs received from each P^j inside the FOV of P^i.

To make this algorithm even more scalable, the neural network of a predator P^i has been designed to be recurrent with a fixed number of input neurons. Therefore, each neural network has only 3 input neurons (X^j, Y^j, F^j): X^j and Y^j represent the coordinates and F^j the Compressed Fitness Value for predator P^j. The neural network is expected to sequentially receive a triplet of inputs from each P^j inside the FOV of P^i, and to activate all its neurons once. Finally, the neural network receives the 3 inputs related to P^i, activating and flushing the net to get a final output value α'. α' is treated as a Compressed Fitness Value and an $\lambda' \in [123456789, 987654321]$ is obtained rescaling α'. The first digit of λ' is identified with the cell where the predator should move next. All the steps described up to now belong to the algorithm 1.

In the algorithm 1 the function $compressAllFitnesses(F^i)$ takes the matrix F^i with the 9 fitnesses of the adjacent cells where P^0 could move next, and transforms it into $\alpha \in [-1, 1]$. Then, α is sent to other predators which use it as input to their neural networks. Each other predator P^i uses its neural netwotk to calculate α', which is passed to the inverse function of $compressAllFitnesses$, that is $getNextCellToMove$. $getNextCellToMove(\alpha')$ "decompresses" α', obtaining λ' and returns the first digit of λ', which identifies the best fitted movement to do next.

As long as the neural networks of predators need to be recurrent, an algorithm capable of training this kind of networks is needed. For that, we propose to use Neuroevolution of Augmenting Topologies (NEAT [8]). In our work we have set up populations of predators and tested their performance against a random prey and an evading prey. Predators were rewarded for early catching the prey and punished for each collision between them. Finally, we got the best neural network from the last population and used it within our predators to test if it reduced the collisions and improved performance. Results of this experiment are shown in Sect. 4.

Algorithm 1. Coordinate decisions of predator P^0 with each P^i decision

Require: P^0 = predator with a 3-to-1 recurrent neural network
Require: P = a set of predators inside FOV of P^0
1. Let F^0 be a $3x3$ real matrix
2. Let $\overrightarrow{C}$ be a vector of real numbers
3. Let α, α' be real numbers
4. **for all** $((x, y) \mid 1 \leq x \leq 3, 1 \leq y \leq 3)$ **do**
5. $\quad F^0_{xy} \Leftarrow calculateExtendedKorfFitness(x, y)$
6. **end for**
7. $\alpha \Leftarrow compressAllFitnesses(F^0)$
8. $sendCompressedFitnessToOtherPredators(\alpha, P)$
9. $\overrightarrow{C} \Leftarrow receiveCompressedFitnessesOfPredators(P)$
10. **for all** $(P^i \in \{P \setminus P^0\})$ **do**
11. $\quad \alpha \Leftarrow getCompressedFitnessOf(P^i, \overrightarrow{C})$
12. $\quad (x, y) \Leftarrow getLocationOfPredator(P^i)$
13. $\quad activateNeuralNetworkWithValues(x, y, \alpha)$
14. **end for**
15. $\alpha \Leftarrow getCompressedFitnessOf(P^0, \overrightarrow{C})$
16. $(x, y) \Leftarrow getLocationOfPredator(P^0)$
17. $\alpha' \Leftarrow flushNeuralNetworkWithValues(x, y, \alpha)$
18. $c \Leftarrow getNextCellToMoveTo(\alpha')$
19. $movePredatorTo(P^0, c)$

4 Results

To validate our approach we compared the results of the 3 methods (ExtKorf, ExtKorf+CSN and ExtKorf+CSN+NECool) between them and against the original of Korf in the same environment conditions, but variating the FOV. We measured predators against two different preys: a random moving prey, and a evading prey. The second prey moved to the adjacent cell that is more distant from the closest predator. These tests let us show the magnitude of the improvements and their relative relevance. For running the simulations we used Kok and Vlassis' Pursuit Domain Package (PDP) [9]. Concretely, we used a 30×30 cells field, allowing agents to move diagonal, with the prey starting on the center and predators starting randomly placed. We lauched 4 predators and 1 prey, and to capture the prey predators needed to be surrounding the prey orthogonally, without touching it. In case of collision, predators colliding were penalized being replaced randomly. Simulations were always ran for 500 consecutive episodes.

Our first experiment measured relative improvement of the Extended Korf algorithm against original Korf's. Results shown an improvement of an order of magnitude in most cases. The improvement was much greater when the FOV was unlimited. Namely, for the maximum FOV (15 cells), relative improvement of ExtKorf against Korf was of $\frac{50}{490} \rightarrow 90\%$, while for the minimum FOV (3 cells) it was $\frac{1049}{3007} \rightarrow 65\%$. It is normal, though, that the maximum improvement in cycles and collisions happened when agents could sense the whole world at once. In this case, predators did not lose cycles in trying to find the prey and

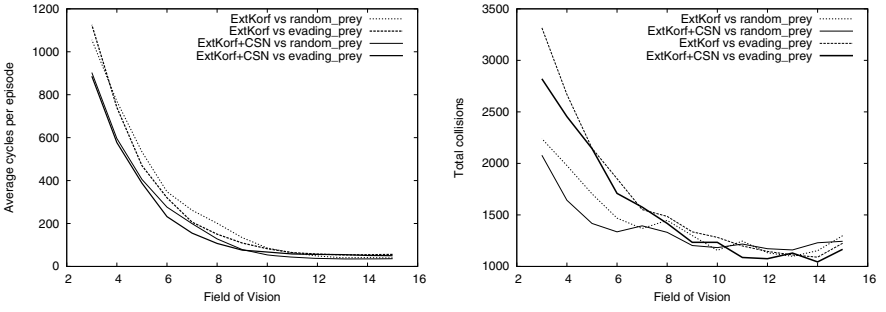


Fig. 3. Comparison between Extended Korf's model and Extended Korf's model with CSN with respect to average cycles per episode and total collisions in 500 episodes

always knew where other predators were. As long as this was the most obvious improvement and for space reasons, we deliverately leave out the graph to center on the other improvements and analyze them more deeply.

As we stated in previous section, there are two major ways of improvement: (1) more efficiently finding the prey and (2) avoiding collisions. We have made two proposals to cover each of these two ways. The CSN protocol enables predators to locate the prey earlier by using the indications from other predators. To check the relative improvement of CSN, we have measured Extended Korf's performance with and without CSN. Figure 3 shows results of this comparison. As expected, results suggest that there are plenty of situations in which CSN saves cycles of exploration to the predators. On average, CSN reduces time spent by predators in finding the prey, and it is more significant when FOV is minimal.

However, CSN is not a definitive solution. CSN turns less effective when dimensions of the world increase due to the necessity of predators to be inside FOV of others to communicate with them. This limits the relative improvement that could be achieved with CSN to a factor depending on the relation of FOV with the size of the world. The less proportion of cells a predator is able to perceive, the more difficult to communicate with others and the more difficult to find the prey.

Altough these results suggest that CSN could be improved, it is not an easy task because FOV restricts communication between agents. Therefore, other way to globally improve performance is to reduce collissions between predators. NECool addressed this issue. To test NECool we set up a training session of 250 generations, with a population of 100 predators. Each predator was tested by 50 episodes against each type of prey, with 6000 cycles as maximum episode time to capture it. The fitness function used to train predator was $f(n, c) = \frac{6001}{n+10c+1}$, which depends on the average number of cycles to capture the prey (n) and the average number of collisions (c). All agents had 6 cells as FOV.

Once we had trained NECool predators, we ran for them the same 500 episodes test we had run earlier, but this time against ExtKorf+CSN predators. The result (see Fig. 4) was a dramatic reduction in the number of total collisions, and this reflected directly in an improvement in the average number of cycles to capture the prey, by around $25 - 35\%$. It is interesting to notice that predators

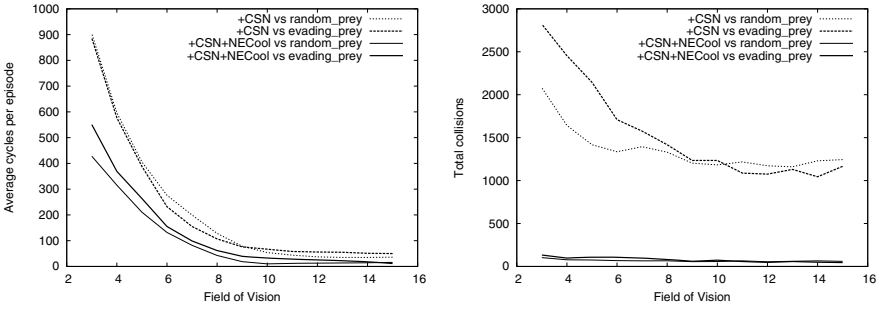


Fig. 4. Comparison between Extended Korf's model with CSN and with CSN+NECool with respect to average cycles per episode and total collisions in 500 episodes

were trained with a FOV of 6 cells but tested with different FOVs. To do this, the information passed to the neural network about (x, y) coordinates of each predator inside FOV of P^0 was always normalized and used coordinates of P^0 as origin. This way, predator coordinates were always independent and scalable to the real FOV in use. If agents had a FOV of f cells, and a predator P^i was located in coodinates (x, y) relative to P^0, the information passed to the neural network of P^0 about the coordinates of P^i was $(\frac{x}{f}, \frac{y}{f})$. Therefore, predators always used coordinates in range $[-1, 1]$, whichever the FOV was.

5 Conclusions and Further Work

Many authors claim that there are interesting challenges to face in the pursuit domain environment. This paper considers coordination problems that arise in some interesting instances of this environment. It proposes a way to mix the efficiency of greedy approaches with the power of evolutive algorithms to get the best of both. The proposal starts extending the approach of Korf (ExtKorf) and continues adding two cooperative strategies: Cascading Sight Notice (CSN) and NEAT Coordination Protocol (NECool).

In order to validate the proposal, the algorithms ExtKorf, ExtKorf + CSN and ExtKorf+CSN+NECool have been measured under the same environment conditions and compared pairways and against Korf's. The Pursuit Domain Package has been used to simulate the environment with the desired challenging conditions. A first experiment was ran to compare measured performance in cycles and collisions between Korf and Extended Korf algorithms. Results of this experiment showed an improvement of an order of magnitude in most cases. A second experiment demonstrated that adding CSN to ExtKorf reduces the average number of cycles to capture the prey, but the reduction is only significant under some conditions. This was mainly due to the necessity of predators to be inside FOV of others to communicate with them. A third experiment added NECool and compared it with ExtKorf+CSN. Results showed a dramatic reduction of total collisions between predators, which mapped directly to a significant improvement (25 to 35%) in average number of cycles to capture the prey.

In consequence of what experiments have shown, we conclude that mixing greedy and evolutive approaches is a promising path to explore. Our final algorithm, ExtKorf+CSN+NECool achieved great results mainly due to its ability to make predators collaborate in an efficient way to lower down collisions with minimum impact in the greedy way to chase the prey. However, there remains room for improvements. For instance, it is still needed a way to early find the prey, what could be achieved if predators coordinate to explore the world, rather than exploring it randomly. Our future work will address this issue and it will also focus on lowering down collisions to 0, with the minimum impact on chasing efficiency.

References

1. Stone, P., Veloso, M.M.: Multiagent systems: A survey from a machine learning perspective. Autonomous Robots 8(3), 345–383 (2000)
2. Benda, M., Jagannathan, V., Dodhiawalla, R.: On optimal cooperation of knowledge sources. Technical Report Tech. Rep. BCS-G2010-28, Boeing AI Center, Boeing Computer Services, Bellevue, WA (1986)
3. Korf, R.E.: A simple solution to pursuit games. In: Proceedings of the 11th International Workshop on Distributed Artificial Intelligence, Glen Arbor, MI, pp. 183–194 (1992)
4. Haynes, T., Sen, S.: Evolving behavioral strategies in predators and prey. In: Sen, S. (ed.) IJCAI 1995 Workshop on Adaptation and Learning in Multiagent Systems, Montreal, Quebec, Canada, pp. 32–37. Morgan Kaufmann, San Francisco (1995)
5. Tan, M.: Multi-agent reinforcement learning: Independent vs. cooperative learning. In: Huhns, M.N., Singh, M.P. (eds.) Readings in Agents, pp. 487–494. Morgan Kaufmann, San Francisco (1997)
6. Jim, K.C., Giles, C.L.: Talking helps: Evolving communicating agents for the predator-prey pursuit problem. Artificial Life 6(3), 237–254 (2000)
7. Katayama, K., Koshiishi, T., Narihisa, H.: Reinforcement learning agents with primary knowledge designed by analytic hierarchy process. In: SAC 2005: Proceedings of the 2005 ACM symposium on Applied computing, pp. 14–21. ACM, New York (2005)
8. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation 10(2), 99–127 (2002)
9. Kok, J.R., Vlassis, N.: The pursuit domain package. Technical Report Technical Report IAS-UVA-03-03, Informatics Institute, University of Amsterdam, The Netherlands (2003)

Multiagent-Based Portfolio Simulation Using Neural Networks

Darius Plikynas

Vilnius Management School, IT department,
J. Basanaviciaus 29a, Vilnius, Lithuania
darius.plikynas@vva.lt

Abstract. The focus of this study is on creating a novel platform for portfolio simulations using neural networks, which are uniquely designed to imitate various well known investing strategies. The artificial investing agents are not only forecasting but also managing their portfolios. The proposed multi-layered framework constructs (i) an intelligent agent composed of a few trained neural networks, each specialized to make investment decisions according to the assigned investment strategy, (ii) multi-agent systems, which are trained to follow different investing strategies in order to optimize their portfolios. The novel multiagent-based portfolio simulation platform gives us an opportunity to display the multi-agent competitive system, and find the best profit-risk performing agents, i.e. investing strategies. In sum, simulations show that the proposed NN-based multi-agent system produces similar results in terms of e.g. wealth distribution compared with empirical evidence from the actual investment markets' behavior as described by Pareto wealth and Levy returns distributions.

Keywords: Neural Networks, Agent-Based Computational Finance, Artificial Stock Markets, Social Simulation.

1 Introduction

Following a rationality assumption, our agent is a subjective (expected) utility maximizer. We hold this assumption to be true, when designing our model. The level of savings, the portfolio and risk aversion that agents choose, may be affected by these individual factors. One way to think of this is to think of agents who have the same objective, but different expectations, and therefore, hold different portfolios. This will cause wealth to flow between the agents over time [1], [10].

In the economy in which agents have heterogeneous beliefs or heterogeneous learning rules, structure on long-run prices arises from the forces of market selection. It has long been assumed that those with better beliefs will make better decisions, driving out those with worse beliefs, and thus determining long run assets prices and their own welfare [12], [5].

In the current approach, we hold that a very effective way to employ a passive strategy with common stocks is to invest in an indexed portfolio. An increasing amount of mutual fund and pensions fund assets can be described as passive equity

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 816–826, 2008.

investments [3]. Using index funds, these asset pools are designed to duplicate as precisely as possible, the performance of some market indexes.

Regarding the methodical aspect of the proposed novel technique (see the following sections), reinforced learning seems to be a close match, as it is a sub-area of machine learning, concerned with how an agent should act in an environment so as to maximize some notion of long-term reward.

Recent extensive analysis of short-term returns has shown that price differences, log-returns, and rates of return are described extremely well by a truncated Levy distribution [1]. In this respect, proposed research is also significant, since it provides a simulation framework within which empirical laws like power-law and Levy distribution can be observed and studied within the parameter's space of the simulated model [9], [12], [10].

In fact, almost all proposed systems are either employed for various technological or robotics applications and very few of them like the Multilayered Multi Agent Situated System (MMASS), Agent-Based Computational Demography (ABCD) or Agent-Based Computational Economics (ACE Trading World application: simulation of economic phenomena as complex dynamic systems using large numbers of economic agents involved in distributed local interactions [14]) are suitable for programmable simulations of social phenomena.

The full functionality of the proposed framework embraces these five main components: 1) resources (information and natural resources), 2) wavelike transformation (WT), 3) neural networks (NN), 4) evolutionary computing (EC), 5) production rules (PR). For the current research stage, though, we have only designed and optimized the neural network component and simulation environment (http://vlab.vva.lt/; Investment Simulation_v1). However, even during design of the first stage, we had to foresee what is coming next. Therefore, we have built, in advance, a more sophisticated simulation framework than is needed for the presented application below.

The paper is organized as follows. Section 2 gives a brief understanding of the application domain, whereas, section 3 addresses the experimental setup, giving us more details about the design and implementation of the proposed approach. Besides, it offers an overview of operational procedures, i.e. how the agent operates in the investment environment. Section 4 summarizes the findings and discusses further implications.

2 Application Domain

According to the above discussion, this section discusses research issues related with the investment market, which is represented using a real data set from S&P500, Dow Jones Composite and US Treasury Bonds indices (value, open/close values, trade volumes, volatility etc). Each agent, according to his investment strategy, is optimized for the best possible forecasting of index fund fluctuations. This system of agents is targeted to find an optimal risk-profit assessment on the efficient frontier according to Markowitz theory [8].

The procedure of building a portfolio of financial assets is following certain steps. Specifically it identifies optimal risk-return combinations available from the set of index funds and non-risky assets, following Markowitz efficient frontier analysis.

In the proposed model, each agent is forming his portfolio composed from the three investment options (in the next version the number of portfolio assets will not be bound to only three assets): two index funds and a Treasury bond, see above. This is a minimum set of assets needed for the design of a profit-risk adjusted portfolio, where the Treasury bond is assumed as an almost risk free asset.

It is important to notice that during simulation, an agent can change one owned asset to another, having associated transaction costs, but no liquidity issues. The transfer of an owned asset is via the current selling market price, and the buying of another asset is via the current market buying price. Therefore, there is no need to design any auctioning mechanism, and address liquidity issues.

If our simulations are showing that proposed NN-based multi-agent system produces similar results in terms of wealth distribution comparing with empirical evidences from the real investment markets' behavior, as described by Pareto wealth and Levy returns distributions [6], then we can conclude, that the proposed model is close to reality.

So, Pareto discovered that at the high-wealth range (social agent's wealth is in the high-wealth range if it is over some typical value characteristic for that particular society. In the current case, W_0 was chosen as 20% of the wealthiest out of all population), wealth (and also income), are distributed according to a power-law distribution. This law became known as the Pareto distribution or Pareto law. The Pareto distribution is given by the following probability density function

$$P(W) = CW^{-(1+\alpha)} \qquad for\ W \geq W_0, \tag{1}$$

where W stands for wealth, $P(W)$ is the density function, W_0 is the lower of the high-wealth range, C is a normalization constant, and α is known as the Pareto constant. Pareto's finding has been shown in numerous studies to provide an excellent fit to empirical wealth distribution in various countries [1].

Pareto distribution is related to wealth distribution, which is acquired by the cumulative process of price differences. Meanwhile, distribution of stock returns, especially when calculated for short time intervals, are not suited to the normal distribution. Rather, the distributions of stock returns are leptokurtic or "fat tailed". Mandelbrot [6] has suggested that log-returns are distributed according to the symmetrical Levy probability distribution defined by

$$L^{\gamma}_{\alpha_L}(x) \equiv \frac{1}{\pi}\int_0^{\infty} \exp(-\gamma \Delta t q^{\alpha_L})\cos(qx)dq \tag{2}$$

Above, is estimated Levy probability density function at x; α_L is the characteristic exponent of the distribution; $\gamma\Delta t$ is a general scale factor; γ is the scale factor for $\Delta t=1$, and q is an integration variable [7]. Recent analysis of returns has shown that price differences, log-returns, and rates of return are described extremely well by a truncated Levy distribution [5], [6].

The next section outlines how the actual experimental setup is constructed. It outlines the main design and methodical issues, and also explains how the NN-based multi-agent system is configured to meet the intended empirical application goals.

3 Experimental Setup

In the proposed model, each investor is represented as an agent. Depending on the simulation parameter set, the total number of agents can be chosen, varying from a few to even thousands. In fact, one could simulate as many agents as possible in order to get power law distributions of their portfolios, but we are not intended to simulate the behavior of the whole market. Asset prices are given as exogenous factors, i.e. our subsystem of agents composes just a small fraction in the overall market, which doesn't significantly influence the overall market behavior and asset prices respectively.

In a methodical sense, agents are modeled using multilayer perceptron (MLP) backpropagation neural networks [NN] with fast Levenberg-Marquardt learning algorithm [4]. The application is compiled using MatLab environment and NN Toolbox in particular. To summarize, the virtual lab software platform is based on Condor, Tomcat server and Oracle XE DB. The Win2003 server is running on Intel Core2 Quad Q6600 2.4 GHz/1066FSB/8MB/RAID2*500GB HDD hardware server platform (integrated 4 processors).

Concerning the "internals" of an agent, each investment asset is analyzed by a dedicated NN (neural network). This means, that an agent is a modular entity composed from a number of different NNs, where separate NNs are dealing with different market data, e.g. time series analyses, forecasting, recognition (when and which investment strategy to employ) etc.

For the effective implementation of the model there is an additional algorithmic layer of procedures above NNs. It deals with agents' portfolios and operational decisions. Thus, profitable decision-making is not solely based on the NNs performance alone (see Fig.1). Therefore, the second layer assists as procedural shell, which maintains market interventions in the operational sense, and also feedback, and agent's own capital estimates.

The employed connectionist approach is working rather like a reactive system, implementing direct mappings from perception to action that avoid expensive intermediate steps of representation and reasoning. This loosely follows the idea that intelligent systems should be constructed from large numbers of separate agents, each with percepts, actions, and goals of its own.

Now let's focus on the data setup. All real market data is collected from the publicly available yahoo/finance data source, where we got our S&P500, Dow Jones Composite and US Treasury Bonds indices data sets (opening, closing, volume, average price, etc). This data is checked for consistency, but is not de-trended as we definitely wanted all periodic and non-periodic fluctuation to be present for NNs to learn.

The reader should also be aware that the size of the data set plays a crucial part in the NN learning and testing stage (in the current case, over 6581 business day observations were collected beginning from the 1981-ies; 1581 observations were dedicated for the NN testing stage). As a standard procedure, optimally designed NN went through the initial training and validation stage. Only afterwards were they ready for the simulation stage, i.e. forecasting and recognizing, see Fig 2. NN learned to forecast the market behavior (index fluctuations, trade volume, market timing moments, specific patterns described by the technical analyses etc) and recognize investment strategies.

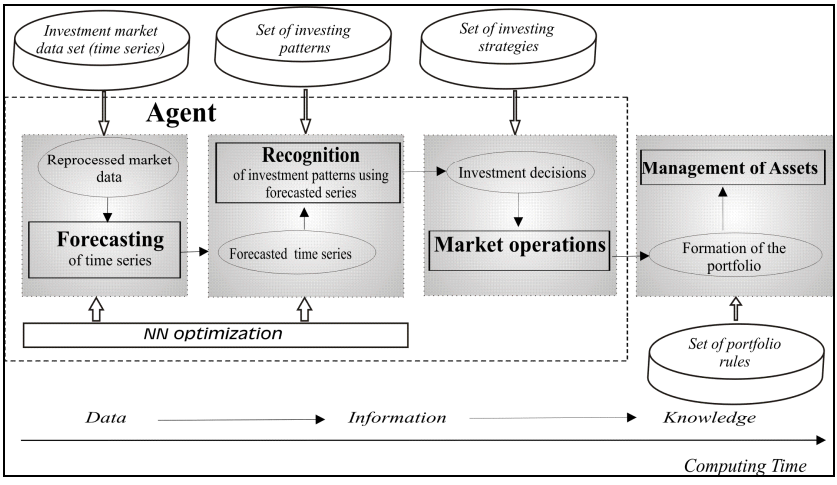


Fig. 1. General simulation model scheme

In general, the more data we have, the better we are in forecasting and recognizing if overfitting of NN is ruled out. In the model, the experimenter can generate a set of NNs with differing internal configurations (learning algorithms, topology, number of epochs, biases, etc), but for simplicity at this stage we were using only one type of optimized NN, which has been tailored for all agents later.

Once we have NN optimized for forecasting of the particular data set, we can make use of it for recognizing different investing strategies by simply teaching it to do so. In the current study, we applied seven investing strategies. Therefore, one and the same NN configuration has been trained with different investing strategies obtaining appropriate NNs, each tailored for a different type of investing strategy (naïv, buy and hold, bull trading, support&resistance, contrary, risk averse, filter rule).

Following the previous discussion, we are investigating how traditional strategies are operating in our multi agent model. As we can see from Fig. 2, all strategies are wrapped in the same principal simulating scheme.

Actually, the stage of forecasting and recognition usually go one after another, though, in this particular model, the NN learns forecasting and recognizing at the same time (the initial data set has been preprocessed to add needed technical indicators for the recognition of strategies). MSE-mean square error and R-square were used mainly as performance criteria, comparing with the standard multivariate regression. As a matter of fact, a novel type of NN has been designed, which has a unique input weight matrix designed [11].

In conclusion, a model was designed that could relate the domain space (real investment market data) with solutions space (simulated agents' investment decisions). A modular NN-based approach was adopted for simulation of complex strategies existing in the investment markets, see Fig. 1. An agent as a backpropagation NN (each NN for an asset) is aimed at providing (i) forecasting of asset values, (ii) recognizing market situations, (iii) making timely market interventions.

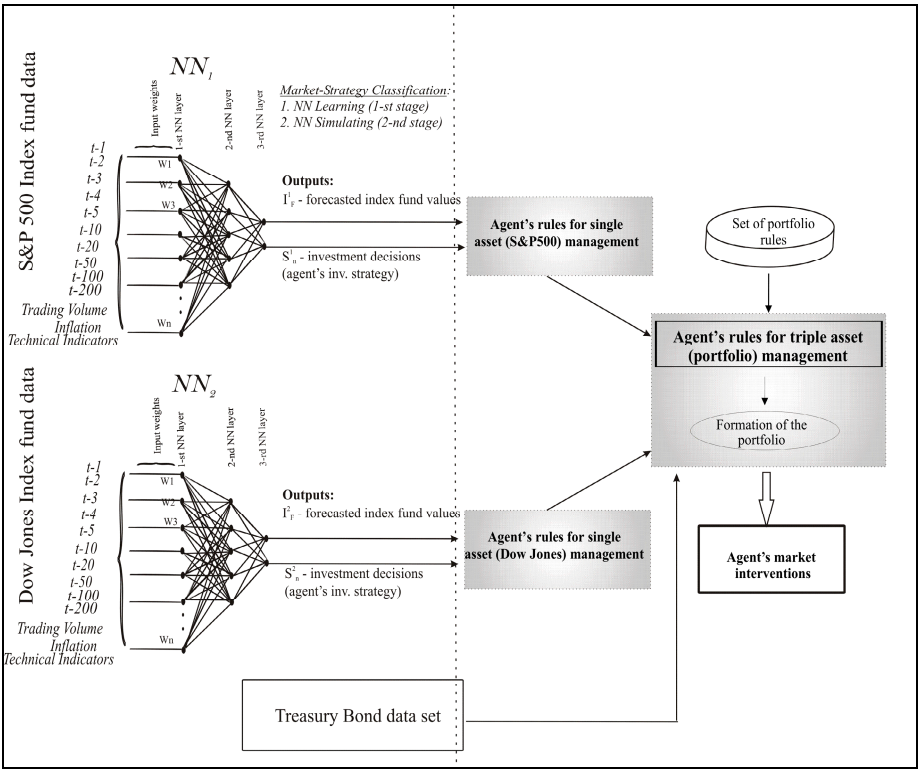


Fig. 2. Agent's principal simulating scheme

The leading experts in the field [3], [2], [13] have published a number of papers providing support for some basic technical indicators, e.g. MACD – moving average convergence/divergence, W%R – Williams' oscillator, MFI – money flow index, WMA – weighted moving average etc. Following their advice, in the presented model, we have adopted them as a benchmark. Therefore, our model addresses not only trend issues, and selling/buying volumes, but also more sophisticated indicators like filtering rules, e.g. "buy a stock if the price moves up 10 percent from some established base, such as a previous low, hold it until it declines 10 percent from its new high, and then sell it".

This adopted approach makes agent's decisions effective in real time, where the agent is able to execute intelligent market timing decisions (interventions). The diversity of investment strategies makes it possible for agents' to execute different investment patterns, following chosen profit/risk targets.

To the author's knowledge, there is no experiment in the field that focuses on the market mechanisms in such a virtual setting. The management of the agents' operational procedures thus gives us an insightful breakthrough, and unique tool for the simulation of investment strategies. Simulations' results are discussed next.

4 Results and Conclusions

Hence, our agents simulate the above mentioned investment strategies as we can see in Fig. 3, where the blue line represents the NN performance during simulation of investment decisions following a particular investment strategy (S_2). Even from the visual impression (see lower testing graphs in Fig 3), it is clear that NNs are more or less following real investors' behaviors, and can be delegated to perform routine trading actions in the market.

The ultimate long-term goal for each agent differs, but in general each of them is looking for the initial capital maximization (initial capital equals 1 for each agent). The model is enriched with some heuristics, based on the nonlinear properties and generalization features of neural networks, which, as numerous studies have shown [5], [11], [4], have systematically outperformed the traditional regression methods so popular in finance.

What is next? Now we are ready to go into the real market and test our simulated investment strategies, see Fig. 4. We generate a chosen number of agents for each

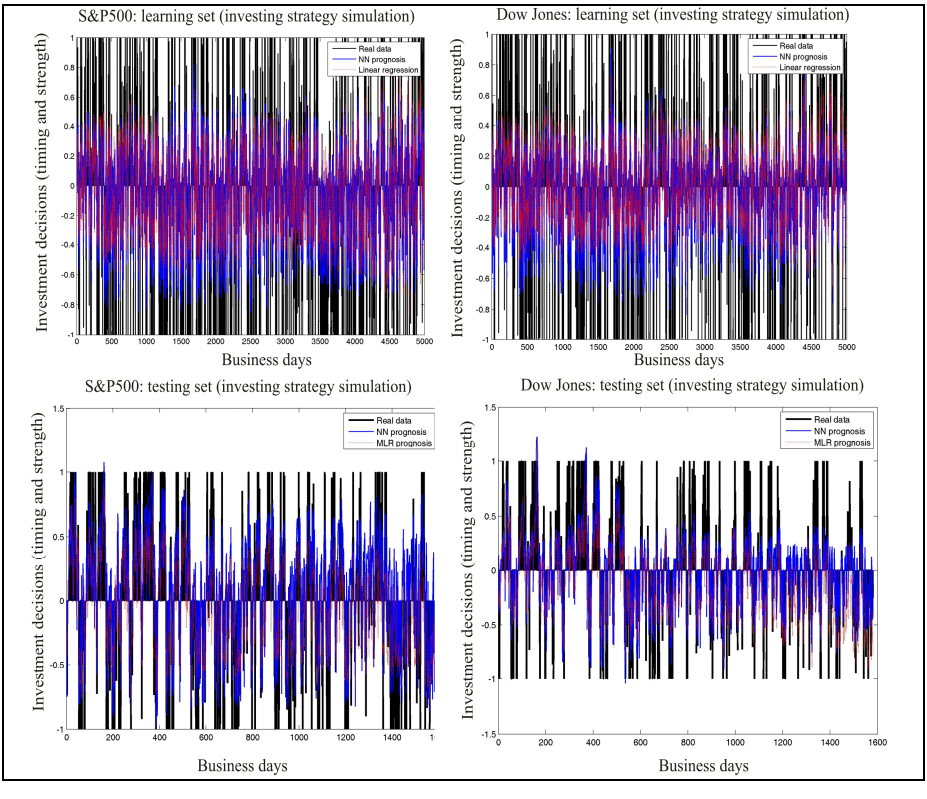


Fig. 3. Investment strategy (S_2) simulation during the NN learning and testing stages (agent's decisions vary between [-1;1], where 1 means all funds reallocation to this specific asset, and -1 means selling all funds allocated in this specific asset)

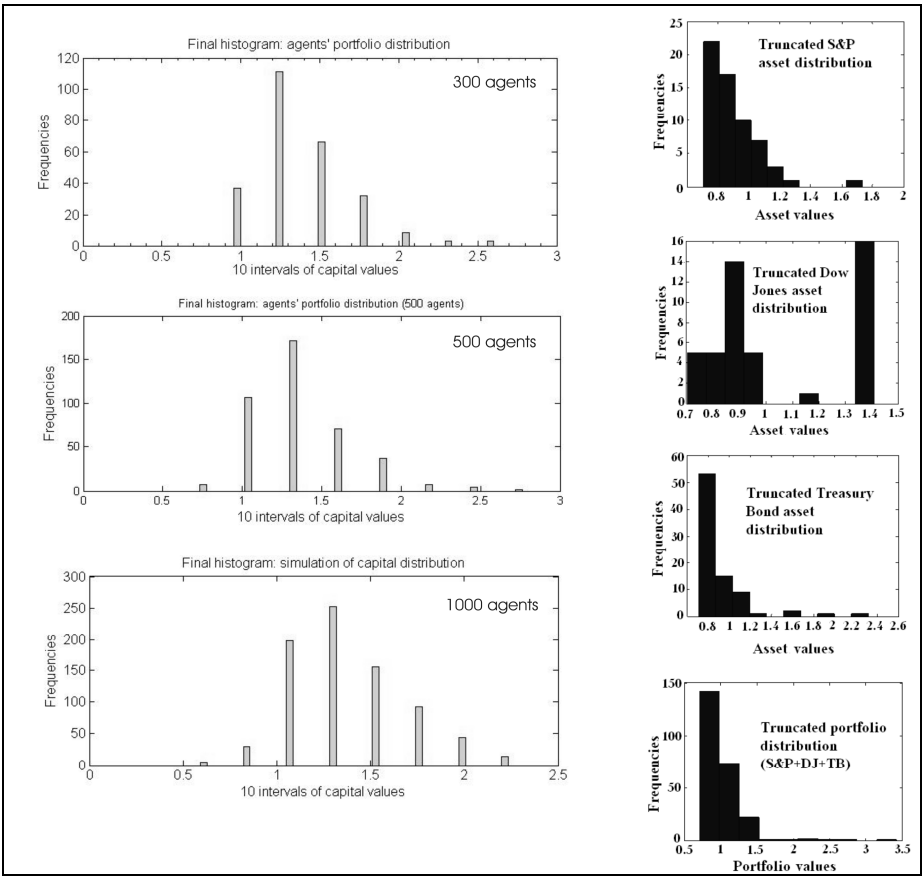


Fig. 4. Simulated wealth distribution (histogram) for strategies $S_{2\text{-}6}$ having 1) different number of investing agents (see left side), 2) truncated asset distribution as required by Pareto distribution law (see right side). Frequencies represent a number of agents with accumulated capital ranging in the 10 intervals of possible values.

investment strategy, allocate them initial capital (initial capital is shared among the three assets proportionally 0.33:0.33:0.33) and start simulation.

Certainly, before doing that we have to parameterize the simulation model itself. After parameters are set and simulation has started, agents are reallocating their funds, i.e. buying and selling assets at the market price (there are no liquidity considerations involved). In this way, they are designing their own portfolio. The final goal is to find out which particular strategies have led to the best performance in terms of accumulated wealth and risk performance.

Empirical tests, whether simulated log-returns are distributed according to the symmetrical Levy probability distribution defined by Eq. 2 or truncated Levy distribution [5], [6], gave a clear answer, i.e. log-returns are described extremely well by the truncated Levy distribution. Though, that is more obvious for portfolio than single

asset returns. Empirical results also gave an insight which simulated strategies are closer to the real investment market behavior.

Consequently, aggregated returns estimated for all agents give us portfolio capital distributions, as it is depicted in Fig. 4 (see left side). According to numerical estimates, which are given below, it perfectly resembles real market behavior. Besides, there are shown simulation results for each asset composing the portfolio, see Fig. 4 (right side).

As we see from the Fig. 4, starting simulation from the initial capital 1, at the end of simulation we have peak frequencies in the range 1.36±0.3 (conf. int. 95%) of initial portfolio value, which well matches with the real market participants' performance [1-3].

In fact, simulated portfolios are following Pareto wealth distribution law (see Eq. 1). There are some agents who perform much better having portfolio values well above the average. The ability of the simulating model to produce such distributions is of particular importance as it clearly validates feasibility of the proposed approach.

For the quantitative reasoning we adapt truncated high-wealth Pareto power law distribution, where $w_0 = 0.7$ (the minimum limit for the "high-wealth" end of distribution), constant $C = \alpha\, w_0^{\alpha}$ (because probability density integral equals 1). We assume, that obtained portfolios are distributed according to the Pareto law, if simulated distribution F_D matches Pareto distribution $F_D \sim Pareto(C, w_0)$. The null hypothesis:

$$\begin{cases} H_0 : F_D = F_{Pareto} \\ H_1 : F_D \neq F_{Pareto} \end{cases}, \tag{3}$$

where F_D – truncated (simulated) distribution function (values with "low-wealth" < w_0 are filtered out). We adapt Kolmogorov-Smirnov test with standard significance level (5%)

There is one more important factor. In reality, ordinary agents' operations like buying/selling are associated with so called transaction costs calculated as percentage from the transaction sum. For the empirical study described above transaction costs were not included, but proposed simulation approach does have this opportunity.

We can impose arbitrary transaction cost rate for each market operation. Hence, the implementation of transaction cost completely changes the profitability levels and wealth distribution accordingly. In sum, research shows, that even very low transaction rates (>0.3% market deal price) significantly reduce investors' wealth if high frequency market interventions are involved. There is an obvious trade-off between market interventions rate and transaction costs: the higher transaction costs, the lower number of interventions are needed to sustain the same wealth level.

From the results above, see Table 1, we can infer that simulated portfolio distribution fits truncated Pareto power law distribution [1], [3], [5], [6]. Though, the truncated portfolio data set is relatively small and p values are not very high. Therefore, for stronger evidence we need at least 10,000 or bigger portfolio set.

The main question addressed by practitioners is, whether particular investment strategy is dealing well in the particular market. They are not only interested in terms of returns (profits), but also in terms of associated risk, as it is described by CAPM. Proposed simulation environment gives such an opportunity. It estimates risks (MSE, VAR, CVAR etc) associated with each investment strategy, as an example.

Table 1. Null hypothesis: testing whether obtained portfolios are distributed according to the Pareto law (see Fig. 4)

Index*	P-value	Null hypothesis
SP	0.937±0.062	H0
DJ	0.767±0.191	H0
TB	0.985±0.003	H0
SP+DJ+TB	0.552±0.250	H0

* - SP, DJ and TB stand for S&P500, Dow Jones Composite and Treasury Bonds (30 year) accordingly; whereas SP+DJ+TB – portfolio values.

For instance, simulation model clearly demonstrates how well each investment agent is doing in terms of risk-profit performance (blue dots represent individual agents). An investment analyst may clearly see the difference between different investment strategies and make probabilistic inference about outcomes. It gives a unique chance to test investment strategies before applying them in the real markets.

The final empirical findings are depicted in the Fig. 5. Here are represented performances of simulated agents (investment strategies) - all portfolios at once. Except couple outliers, the overall simulated portfolio distribution fits very well to the real market data according to the CAPM theory.

Null hypothesis H_0 (whether obtained portfolios are distributed according to the Pareto law, see Eq. 5) is positive then $W_0 = 0.95$ and $\sigma = 3.17$ (p-value 0.91). Notwithstanding obtained results, additional research needs to be done to examine more samples of investment strategies and bigger set of portfolios accordingly.

Simulation gives us an opportunity to display the multi-agent competitive system, and find the best profit-risk performing agents, i.e. investing strategies. This gives an edge over costly and time consuming manual experimentation in the diversified investment markets with real-life index funds (or stocks) and non-risky financial assets. Unfortunately, the limited space available for this paper does not allow to demonstrate plenty of validation and robustness results.

The first version of the model is already available via the web, using our virtual lab environment, which is intended to enable others to access the model and experiment with it (http://vlab.vva.lt/; Investment Simulation_v1).

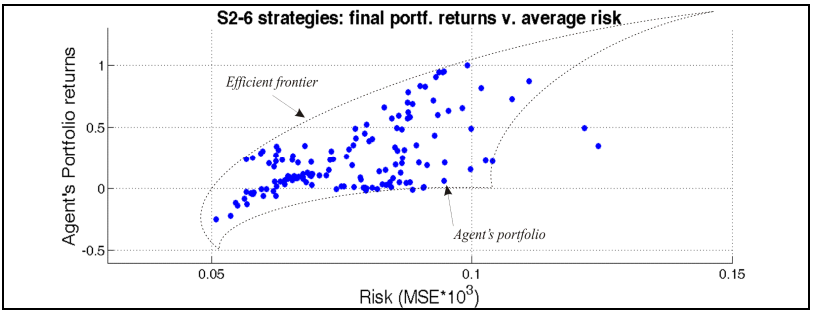


Fig. 5. Final simulated multi-agent system performance (all investment strategies are included): profit-risk based distribution of agents' portfolios

In sum, the modular structure of the system of specialized NN gives us an advantage over the straightforward algorithmic approaches, as it improves forecasting, it is more flexible, and, being nondeterministic, mimics the complexity in its' own terms.

References

1. Blume, L.E., Durlauf, S.N. (eds.): The Economy as an Evolving Complex System III. A Volume in the Santa Fe Institute in the sciences of complexity. Oxford University Press, New York (2006)
2. Darley, V., Outkin, A.V.: A NASDAQ Market Simulation: Insights on a Major Market from the Science of Complex Adaptive Systems, p. 152. World Scientific, Singapore (2007)
3. Jones, C.P.: Investments: Analysis and Management, 8th edn. John Wiley & Sons, Chichester (2002)
4. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, New Jersey (1999)
5. Hoffmann, A.O.I., Jager, W., Von Eije, J.H.: Social Simulation of Stock Markets: Taking It to the Next Level. Journal of Artificial Societies and Social Simulation (JASSS) 10(2,7) (2007), http://jasss.soc.surrey.ac.uk/10/2/7.html
6. Levy, M., Solomon, S.: New Evidence for the Power-Law Distribution of Wealth. Physica A 242 (1997)
7. Mantegna, R.N., Staney, H.E.: An Introduction to Econophysics: Correlations and Complexity in Finance. Cambridge University Press, Cambridge (2000)
8. Markowitz, H.M.: Portfolio Selection: Efficient diversification of investments. Blackwell Publishers, Malden (1991)
9. McCauley, J.L.: Dynamics of Markets: Econophysics and Finance. Cambridge University Press, Cambridge (2004)
10. Moss, S., Edmonds, B.: Towards Good Social Science. Journal of Artificial Societies and Social Simulation 8(4) (2005), http://jasss.soc.surrey.ac.uk/8/4/13.html
11. Plikynas, D.: Proceedings of the First World Congress on Social Simulation, WCSS 2006, August 21-25. Kyoto University, Kyoto (2006)
12. Sapienza, M.: Do Real Options perform better than Net Present Value? Testing in an artificial financial market. Journal of Artificial Societies and Social Simulation 6(3) (2003), http://jasss.soc.surrey.ac.uk/6/3/4.html
13. Sharpe, W.F.: Capital asset prices: A theory of market equilibrium under conditions of risk. Journal of Finance 19(3), 425–442 (1964)
14. Testafsion, L., Judd, K.L.: Handbook of computational economics: agent- based computational economics, vol. 2, p. 828. North Holland, Amsterdam (2006)
15. Wooldridge, M.: Multiagent Systems, p. 346. John Wiley & Sons Ltd., Chichester (2002)

Managing Conflicting Beliefs with Fuzzy Trust on the Semantic Web

Miklos Nagy[1], Maria Vargas-Vera[2], and Enrico Motta[1]

[1] Knowledge Media Institute (KMi)
The Open University
Walton Hall, Milton Keynes
MK7 6AA, United Kingdom
mn2336@student.open.ac.uk, e.motta@open.ac.uk
[2] Computing Department
The Open University
Walton Hall, Milton Keynes
MK7 6AA, United Kingdom
m.vargas-vera@open.ac.uk

Abstract. Automated ontology mapping approaches often combine similarity measures in order to increase the quality of the proposed mappings. When the mapping process of human experts is modeled with software agents that assess similarities, it can lead to situations where the beliefs in the assessed similarities becomes contradicting. The combination of these contradicting beliefs can easily worsen the mapping precision and recall, which leads to poor performance of any ontology mapping algorithm. Typically mapping algorithms, which use different similarities and combine them into a more reliable and coherent view can easily become unreliable when these contradictions are not managed effectively between the different sources. In this paper we propose a solution based on the fuzzy voting model for managing such situations by introducing trust and voting between software agents that resolve contradicting beliefs in the assessed similarities.

1 Introduction

Assessing the performance and quality of different ontology mapping algorithms, which operate in the Semantic Web environment has gradually been evolved during the recent years. One remarkable effort is the Ontology Alignment Evaluation Initiative[1], which provides a possibility to evaluate and compare the mapping quality of different systems. However it also points out the difficulty of evaluating ontologies with large number of concepts i.e. the library track where due to the size of the vocabulary only a sample evaluation is carried out by a number of domain experts. Once each expert has assessed the correctness of the sampled mappings their assessment is discussed and they produce a final assessment, which reflects their collective judgment. Our ontology mapping algorithm

[1] http://oaei.ontologymatching.org/

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 827–837, 2008.

DSSim [1] tries to mimic the before mentioned process, using different software agents as experts to evaluate and use beliefs over similarities of different concepts in the source ontologies. Our mapping agents use WordNet as background knowledge to create a conceptual context for the words that are extracted from the ontologies and employ different syntactic and semantic similarities to create their subjective beliefs over the correctness of the mapping. DSSim addresses the uncertain nature of the ontology mapping by considering different similarity measures as subjective probability for the correctness of the mapping. It employs the Dempster Shafer theory of evidence in order to create and combine beliefs that has been produced by the different similarity algorithms. For the detailed description of the DSSim algorithm one can refer to [2]. Using belief combination has their advantages compared to other combination methods. However the belief combination has received a verifiable criticism from the research community. There is a problem with the belief combination if the agents has conflicting beliefs over the solution. In this paper we present a novel trust management approach for resolving conflict between beliefs in similarities, which is the core component of the DSSim ontology mapping system.

The paper is organized as follows. Section 2 provides the description of the problem and its context. Section 3 describes the voting model and how it is applied for determining trust during the ontology mapping. Section 4 details the example scenario, which is used to demonstrate our proposed solution. In section 5 we present our experiments that have been carried out with the benchmarks of the Ontology Alignment Initiative. Section 6 gives and overview of the related work. Finally, section 7 describes our future work.

2 Problem Description

In the context of the Semantic Web trust can have different meaning therefore before we describe the problem let us define the basic notions of our argument.

Definition 1. *Trust: One mapping agent's measurable belief in the competence of the other agents' belief over the established similarities.*

Definition 2. *Content related trust: Dynamic trust measure that is dependent on the actual vocabulary of the mappings, which has been extracted from the ontologies and can change from mapping to mapping.*

Definition 3. *Belief: The state in which a software agent holds a proposition or premise over a possible mapping of selected concept pair combination to be true.*

Detecting and managing conflicting evidence over the possible meaning of the Semantic Web data is not always obvious. In practice the degree of conflict can differ considerably from case to case. In some cases there can be a considerable difference so the detection is straightforward but in most of the cases it might be difficult to determine which belief is in conflict with the others. Assume that

Table 1. Belief conflict detection

Conflict detection	Belief 1	Belief 2	Belief 3
Obvious	0.85	0.80	0.1
Difficult	0.85	0.65	0.45

we have three software agents, which have carried out similarity assessment on a concept pair mapping. The three agents have used WordNet as background knowledge and build their beliefs considering different concepts context, which was derived from the background knowledge e.g. agent 1 used the direct hypernyms, agent 2 the sister terms and agent 3 the inherited hypernyms. Based on string similarity measures a numerical belief value is calculated, which represent a strength of the confidence that the two terms are related to each other. The scenario is depicted in Table 1.

The values given in Table 1 are demonstrative numbers just for the purpose of providing an example. In our ontology mapping framework DSSim, the similarities are considered as subjective beliefs, which is represented by belief mass functions that can be combined using the Dempster's combination rule. This subjective belief is the outcome of a similarity algorithm, which is applied by a software agent for creating mapping between two concepts in different ontologies. In our ontology mapping framework different agents assess similarities and their beliefs on the similarities need to be combined into a more coherent result. However these individual beliefs in practice are often conflicting. A conflict between two beliefs in Dempster-Shafer theory can be interpreted qualitatively as one source strongly supports one hypothesis and the other strongly supports another hypothesis, where the two hypotheses are not compatible. In this scenario applying Dempster's combination rule to conflicting beliefs can lead to an almost impossible choice, because the combination rule strongly emphasizes the agreement between multiple sources and ignores all the conflicting evidences.

We argue that the problem of contradictions can only be handled from case to case by introducing trust for the similarity measures, which is applied only for the selected mapping and can change from mapping to mapping during the process depending on the available evidences. We propose evaluating trust in the different beliefs that does not depend on the credentials of the ontology owner but it purely represents the trust in a proposed subjective belief that has been established by using different similarity algorithms.

3 Trust Management for Conflicting Belief Combination

In ontology mapping the conflicting results of the different beliefs in similarity can be resolved if the mapping algorithm can produce an agreed solution, even though the individual opinions about the available alternatives may vary. We propose a solution for reaching this agreement by evaluating trust between established beliefs through voting, which is a general method of reconciling differences. Voting is a mechanism where the opinions from a set of votes are evaluated in

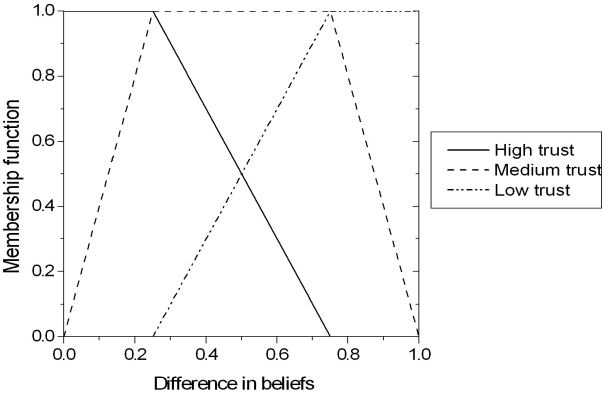


Fig. 1. Trust representation

order to select the alternatives that best represent the collective preferences. Unfortunately deriving binary trust like trustful or not trustful from the difference of belief functions is not so straightforward since the different voters express their opinion as subjective probability over the similarities. For a particular mapping this always involves a certain degree of vagueness hence the threshold between the trust and distrust cannot be set definitely for all cases that can occur during the process. Additionally there is no clear transition between characterising a particular belief highly or less trustful. Therefore our argument is that the trust membership value, which is expressed by different voters can be modeled properly by using fuzzy representation as depicted on Fig. 1. Before each mapping agent evaluates the trust in other agent's belief over the correctness of the mapping it calculates the difference between its own and the other agent's belief. Depending on the difference it can choose the available trust levels. For example if the difference in beliefs is 0.2 then the available trust level can be high and medium. We model these trust levels as fuzzy membership functions.

In fuzzy logic the membership function $\mu(x)$ is defined on the universe of discourse U and represents a particular input value as a member of the fuzzy set i.e. $\mu(x)$ is a curve that defines how each point in the U is mapped to a membership value (or degree of membership) between 0 and 1.

For representing trust in beliefs over similarities we have defined three overlapping trapezoidal membership functions, which represent high, medium and low trust in the beliefs over concept and property similarities in our ontology mapping system. Each voter has similar membership functions but the vertexes of these functions differ from agent to agent.

3.1 Fuzzy Voting Model

The fuzzy voting model was developed by Baldwin [3] and has been used in Fuzzy logic applications. However, to our knowledge it has not been introduced in the context of trust management on the Semantic Web. In this section, we

will briefly introduce the fuzzy voting model theory using a simple example of 10 voting agents voting against or in favour of the trustfulness of an another agent's belief over the correctness of mapping. In our ontology mapping framework each mapping agent can request a number of voting agents to help assessing how trustful the other mapping agent's belief is.

According to Baldwin [3] a linguistic variable is a quintuple $(L, T(L), U, G, \mu)$ in which L is the name of the variable, $T(L)$ is the term set of labels or words (i.e. the linguistic values), U is a universe of discourse, G is a syntactic rule and μ is a semantic rule or membership function. We also assume for this work that G corresponds to a null syntactic rule so that $T(L)$ consists of a finite set of words. A formalization of the fuzzy voting model can be found in [4].

Consider the set of words { Low_trust (L_t), Medium_trust (M_t) and High_trust (H_t) } as labels of a linguistic variable trust with values in $U = [0, 1]$. Given a set "m" of voters where each voter is asked to provide the subset of words from the finite set $T(L)$, which are appropriate as labels for the value u. The membership value $\chi_{\mu_(w)(u)}$ is taking the proportion of voters who include u in their set of labels which is represented by w.

We need to introduce more opinions to the system i.e. we need to add the opinion of the other agents in order to vote for the best possible outcome. Therefore we assume for the purpose of our example that we have 10 voters (agents). Formally, let us define

$$\begin{aligned} V &= A1, A2, A3, A4, A5, A6, A7, A8, A9, A10 \\ T(L) &= L_t, M_t, H_t \end{aligned} \tag{1}$$

where V represents the voters and $T(L)$ describes the trust levels. The number of voters can differ however assuming 10 voters can ensure that

1. The overlap between the membership functions can proportionally be distributed on the possible scale of the belief difference [0..1]
2. The work load of the voters does not slow the mapping process down

Let us start illustrating the previous ideas with a small example - By definition consider our linguistic variable L as TRUST and $T(L)$ the set of linguistic values as $T(L) = (low\_trust, medium\_trust, high\_trust)$. The universe of discourse is U, which is defined as $U = [0, 1]$. Then, we define the fuzzy sets $\mu(Low\_trust), \mu(Medium\_trust)$ and $\mu(High\_trust)$ for the voters where each voter has different overlapping trapezoidal membership functions as described on Fig 1.

The difference in the membership functions is represented by the different vertices of the trapezoid on Fig. 1 ensures that voters can introduce different opinions as they pick the possible trust levels for the same difference in belief.

The random set L=TRUST is defined by the Table 2. Note that in the table we use a short notation L_t means Low_trust, M_t means Medium_trust and H_t means High_trust. Once the fuzzy sets (membership functions) have been defined the system is ready to assess the trust memberships for the input values. Based on the difference of beliefs in similarities the different voters will select the words

Table 2. Possible values for the voting

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
L_t	L_t	L_t	L_t	L_t	L_t	L_t	L_t	L_t	L_t
M_t	M_t	M_t	M_t	M_t	M_t				
H_t	H_t	H_t							

Table 3. Voting

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
H_t	M_t	L_t	L_t	M_t	M_t	L_t	L_t	L_t	L_t

they view as appropriate for the difference of belief. Assuming that the difference in beliefs(x) is 0.67(one agent's belief over similarities is 0.85 and an another agent's belief is 0.18) the voters will select the labels representing the trust level as described in Table 2.

Note that each voter has its own membership function where the level of overlap is different for each voter. As an example the belief difference 0.67 can represent high, medium and low trust level for the first voter(A1) and it can only represent low trust for the last voter(A10). Then we compute the membership value for each of the elements on set $T(L)$.

$$\chi_{\mu(Low\_trust)(u)} = 1 \tag{2}$$

$$\chi_{\mu(Medium\_trust)(u)} = 0.6 \tag{3}$$

$$\chi_{\mu(High\_trust)(u)} = 0.3 \tag{4}$$

and

$$T(L) = \frac{Low\_trust}{1} + \frac{Medium\_trust}{0.6} + \frac{High\_trust}{0.3} \tag{5}$$

A value x(actual belief difference between two agents) is presented and voters randomly pick exactly one word from a finite set to label x as depicted in Table 3. The number of voters will ensure that a realistic overall response will prevail during the process.

Taken as a function of x these probabilities form probability functions. They should therefore satisfy:

$$\sum_{w \in T(L)} Pr(L = w|x) = 1 \tag{6}$$

which gives a probability distribution on words:

$$\sum Pr(L = Low\_trust|x) = 0.6 \tag{7}$$

$$\sum Pr(L = Medium\_trust|x) = 0.3 \tag{8}$$

$$\sum Pr(L = High\_trust|x) = 0.1 \tag{9}$$

As a result of voting we can conclude that given the difference in belief $x = 0.67$ the combination should not consider this belief in the similarity function since based on its difference compared to another beliefs it turns out to be a distrustful assessment. The before mentioned process is then repeated as many times as many different beliefs we have for the similarity i.e. as many as different similarity measures exist in the ontology mapping system.

4 Scenario

Our scenario provides an example for the initial context and the trust evaluation process is detailed in section 3.1. In order to evaluate trust using the fuzzy voting model consider the following scenario:

The mapping algorithm has assessed the beliefs between the concept similarities from two ontologies. The two real word ontologies [2][3] describe BibTeX publications from the University of Maryland, Baltimore County (UMBC) and from the Massachusetts Institute of Technology (MIT).

$$Mapping_{\text{UMBC ontology, MIT ontology}}(publication \leftrightarrow article) \tag{10}$$

where the beliefs in similarities based on different similarity measures are:

$$Agent1_{publication,article} = 0.85 \tag{11}$$

$$Agent2_{publication,article} = 0.75 \tag{12}$$

$$Agent3_{publication,article} = 0.60 \tag{13}$$

There is a conflict between the beliefs, which cannot be determined easily therefore each mapping agent(Agent 1,2,3) has to apply the voting between its own and other agents' beliefs. The voters and their overlapping fuzzy sets are defined as depicted on Fig. 1. Once the voters are ready to assess the trust level of the beliefs, we calculate the belief differences between(Table 4) each mapping agent.

Table 4. Difference in beliefs

	Agent 1	Agent 2	Agent 3
Agent 1	0	0.10	0.25
Agent 2	0.10	0	0.15
Agent 3	0.25	0.15	0

The differences of beliefs in Table 4 serve as the input to the voters. Each voter creates its bet on the trust, which is dependent on the mapping agents' perspective. Each mapping agent considers its own belief as the most trustful and has doubt over the other beliefs. For each mapping agent the trust levels are determined by the voters as described in Section 3.1(Table 2 and 3). The conflicting

[2] http://ebiquity.umbc.edu/ontology/publication.owl
[3] http://visus.mit.edu/bibtex/0.01/bibtex.owl

belief can be determined using fuzzy rules based on the trust results because the mapping agent, which has the conflicting belief will find that two other beliefs are not trusted compared to the real situation where only one belief cannot be trusted.

5 Empirical Evaluation

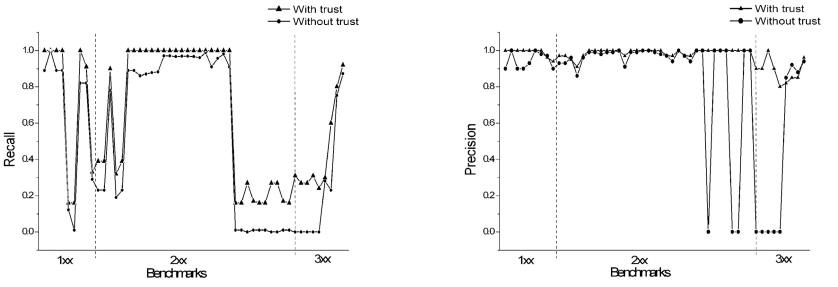
The evaluation was measured with recall and precision, which are useful measures that have a fixed range and meaningful from the mapping point of view. Recall is 100% when every relevant entity is retrieved. Precision is a measure of how well the engine performs in not returning non relevant documents. Precision is 100% when every entity returned to the user is relevant to the query. Both precision and recall has a fixed range: 0.0 to 1.0 (or 0% to 100%). A good mapping algorithm must have a high recall to be acceptable for most applications. The objective is always to make improvement in preferably both in precision and recall.

We have carried out experiments with the benchmark ontologies of the Ontology Alignment Evaluation Initiative(OAEI),[4] which is an international initiative that has been set up for evaluating ontology matching algorithms. The experiments were carried out to assess how trust management influences results of our mapping algorithm. Our main objective was to evaluate the impact of establishing trust before combining beliefs in similarities between concepts and properties in the ontology. The OAEI benchmark contains tests, which were systematically generated starting from some reference ontology and discarding a number of information in order to evaluate how the algorithm behave when this information is lacking. The bibliographic reference ontology (different classifications of publications) contained 33 named classes, 24 object properties, 40 data properties. Further each generated ontology was aligned with the reference ontology. The benchmark tests were created and grouped by the following criteria:

- Group 1xx: simple tests such as comparing the reference ontology with itself, with another irrelevant ontology or the same ontology in its restriction to OWL-Lite
- Group 2xx: systematic tests that were obtained by discarding some features from some reference ontology e.g. name of entities replaced by random strings or synonyms
- Group 3xx: four real-life ontologies of bibliographic references that were found on the web e.g. BibTeX/MIT, BibTeX/UMBC

As a basic comparison we have modified our algorithm (without trust), which does not evaluate trust before conflicting belief combination just combine them using Dempster's combination rule. The recall and precision graphs for the algorithm with trust and without trust over the whole benchmarks are depicted on Fig. 2. Experiments have proved that with establishing trust one can reach higher average precision and recall rate.

[4] http://oaei.ontologymatching.org/



(a) Recall (b) Precision

Fig. 2. Recall and Precision graphs

Figure 2 shows the improvement in recall and precision that we have achieved by applying our trust model for combining contradictory evidences. From the precision point of view the increased recall values have not impacted the results significantly, which is good because the objective is always the improvement of both recall and precision together. We have measured the average improvement for the whole benchmark test set that contains 51 ontologies. Based on the experiments the average recall has increased by 12% and the precision is by 16%. The relative high increase in precision compared to recall is attributed to the fact that in some cases the precision has been increased by 100% as a consequence of a small recall increase of 1%. This is perfectly normal because if the recall increases from 0 to 1% and the returned mappings are all correct (which is possible since the number of mappings are small) then the precision is increases from 0 to 100%. Further the increase in recall and precision greatly varies from test to test. Surprisingly the precision have decreased in some cases(5 out of 51). The maximum decrease in precision was 7% and maximum increase was 100%. The recall have never decreased in any of the tests and the minimum increase was 0.02% whereas the maximum increase was 37%. As mentioned in our scenario in our ontology mapping algorithm there are number of mapping agents that carry out similarity assessments hence create belief mass assignments for the evidence. Before the belief mass function is combined each mapping agent need to calculate dynamically a trust value, which describes how confident the particular mapping agent is about the other mapping agent's assessment. This dynamic trust assessment is based on the fuzzy voting model and depending on its own and other agents' belief mass function. In our ontology mapping framework we assess trust between the mapping agents' beliefs and determine which agent's belief cannot be trusted, rejecting the one, which is as the result of trust assessment become distrustful.

6 Related Work

Dominantly the existing approaches that address the problem of the trustworthiness of the available data on the Semantic Web are reputation and context based

e.g. using digital signatures that would state who the publisher of the ontology is. However on the Semantic Web trust is more than secure communication or digitally signed information. For example the reliability of information about how a similarity algorithm has "understood" the available concept similarities has little to do with secure communication or the organisiation that created the ontology. To date this perspective of trust has not been investigated in the context of ontology mapping. Ongoing research has mainly been focusing on how trust can be modeled in the Semantic Web context [5] where the trust of user's belief in statements supplied by any other user can be represented and combined. Gil and Ratnakar[6] presents an approach that captures the trust of individual users based on an actual context of the source, which determines how to use information from each source. According to Finin and Joshi[7] trust management on the Semantic Web requires a semantic policy language for defining security requirements and a distributed trust management approach, which includes a way of expressing security and belief information in a clear and concise manner so that its meaning is unambiguous. Their solution considers all resources of the Semantic Web e.g. web pages, web services, agents, and human users. Considering multi-agent systems on the Web existing trust management approaches have successfully used fuzzy logic to represent trust between the agents from both individual[8] and community[9] perspective. However the main objective of these solutions is to create a reputation of an agent, which can be considered in future interactions.

7 Conclusion

In this paper we have shown how the fuzzy voting model can be used to evaluate trust and determine which belief is contradictory with other beliefs before combining them into a more coherent state. We have proposed new levels of trust in the context of ontology mapping, which is a prerequisite for any systems that makes use of information available on the Semantic Web. Our system is flexible because the membership functions for the voters can be changed dynamically in order to influence the outputs according to the different similarity measures that can be used in the mapping system. We have described initial experimental results with the benchmarks of the Ontology Alignment Initiative, which demonstrates the effectiveness of our approach through the improved recall and precision rates. There are many areas of ongoing work, with our primary focus being additional experimentation to investigate different kind of membership functions for the different voters and to consider the effect of the changing number of voters and the impact on precision and recall. We also aim to measure the proportion of the obvious and difficult conflicts (see Table 1) that can occur during the mapping process and how these affect the overall performance of our solution.

References

1. Nagy, M., Vargas-Vera, M., Motta, E.: Dssim - managing uncertainty on the semantic web. In: Proceedings of the 2nd International Workshop on Ontology Matching (2007)
2. Nagy, M., Vargas-Vera, M., Motta, E.: Multi-agent ontology mapping with uncertainty on the semantic web. In: Proceedings of the 3rd IEEE International Conference on Intelligent Computer Communication and Processing (2007)
3. Baldwin, J.F.: Mass assignment Fundamentals for computing with words. In: L. Ralescu, A. (ed.) IJCAI-WS 1997. LNCS, vol. 1566, pp. 22–44. Springer, Heidelberg (1999)
4. Lawry, J.: A voting mechanism for fuzzy logic. International Journal of Approximate Reasoning 19, 315–333 (1998)
5. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the semantic web. In: Proceedings of the 2nd International Semantic Web Conference, pp. 351–368 (2003)
6. Gil, Y., Ratnakar, V.: Trusting information sources one citizen at a time. In: Proceedings of the 1st International Semantic Web Conference, pp. 162–176 (2002)
7. Finin, T., Joshi, A.: Agents, turst, and information access on the semantic web. ACM SIGMOD Record, Special section on semantic web and data management 31, 30–35 (2002)
8. Griffiths, N.: A fuzzy approach to reasoning with trust, distrust and insufficient trust. In: Proceedings of the 10th International Workshop on Cooperative Information Agents, pp. 360–374 (2006)
9. Rehak, M., Pechoucek, M., Benda, P., Foltyn, L.: Trust in coalition environment: Fuzzy number approach. In: Proceedings of The 4th International Joint Conference on Autonomous Agents and Multi Agent Systems - Workshop Trust in Agent Societies, pp. 119–131 (2005)

Optimal Agendas and Procedures for N-Issue Negotiation: An Inductive Definition

Saidalavi Kalady[1], B. Dev[1], Arun A.N.[1], V.K. Govindan[1], and Abraham T. Mathew[2]

[1] Department of Computer Science and Engineering
[2] Department of Electrical and Electronics Engineering National Institute of Technology Calicut
said@nitc.ac.in, devbaburaj@gmail.com, arun_4_8@yahoo.com, vkg@nitc.ac.in, atm@nitc.ac.in

Abstract. This paper discusses the multi-issue negotiation between two self-interested agents. The outcome of such negotiations depends on the *agenda* and the negotiation *procedure*. While most of the existing works are concerned with the negotiation strategy, we study the impact of varying agenda and procedure on the negotiation outcome. This is done under incomplete information settings with the equilibrium strategies defined for *issue-by-issue* and *package deal* negotiation procedures. On the basis of these strategies, the *optimal agendaprocedure combination* for each agent is determined.

1 Introduction

Multi Agent Systems (MAS) are fast emerging as an important paradigm which facilitates the modeling and development of large scale distributed systems [6]. The specific properties of MAS like autonomy and self-adaptation make them highly suitable for automated negotiations. Negotiation may be defined as the process by which agents communicate and compromise to reach mutually beneficial agreements [2]. Negotiation can either be over a single issue or on multiple issues. In single issue negotiation, the outcome depends only on the negotiation strategies of the agents. For multi-issue negotiations, in addition to the strategy, the outcome depends on the agenda and negotiation procedure.

Agenda refers to the set of issues under negotiation and the order in which they negotiate [2]. Agenda can be defined either *exogenously,* i.e. before the negotiation begins or *endogenously,* i.e. agents themselves decide during the course of negotiation [1]. Negotiation procedure specifies how the issues will be settled. There are two ways of negotiating multiple issues. One method is to discuss all the issues together which is termed as *package deal.* The other method is to negotiate the issues independent of each other called *issue-by-issue* [3]. This can be done either sequentially or in parallel [2]. The two negotiation procedures can lead to different outcomes.

For a multi-issue negotiation under incomplete information settings, the ideal solution is one that is *Pareto optimal* [4]. A solution is said to be *Pareto optimal* if no agent can be better off without sacrificing the other's utility [5]. An efficient negotiation strategy should thus be able to produce *Pareto optimal* solutions for multi-issue negotiation.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 838–847, 2008.

Most of the existing works on multi-issue negotiations concentrate only on the negotiation strategy, assuming the agenda and procedure to be fixed [4]. However, varying the agenda procedure combinations can have a marked impact on the outcome of the negotiation depending on the scenario under which the negotiation is taking place. Fatima S S et. al. have studied the effect of varying the agenda and procedure in a two issue negotiation model with self interested agents having incomplete information about each other [2]. In the present work, we propose the optimal agenda-procedure combinations for each agent in an *n-issue* negotiation scenario.

This paper first gives an introduction to the negotiation model where we define the negotiation strategy for both *package deal* and *issue-by-issue* negotiation procedures. The strategy defined in this paper is highly abstract as it defines only the actions of the agents without explaining either the *propose* or *concede* strategies. This is because this work concentrates on the impact of agenda/procedure on the outcome, and holds for any strategy. We then explain the agenda-procedure preferences for agents for a single issue and two issue negotiation models. The agent's preference in these situations/models is then used as a base case for providing an inductive definition for the optimal agenda-procedure combination, first for the three issue negotiation model and then for the *n-issue* model.

2 Negotiation Model

Let a and $\bar{a}$ denote two self-interested agents negotiating over n issues A_1, A_2…, A_n. Each agent has complete information about its own parameters but has incomplete information about its opponent's. The information state of each agent is defined as:

$$I^a = \{ RP^a_{A_1} \dots RP^a_{A_n}, T^a, \delta^a, U^a, RP^{\bar{a}}_{A_1} \dots RP^{\bar{a}}_{A_n} \}$$
$$I^{\bar{a}} = \{ RP^{\bar{a}}_{A_1} \dots RP^{\bar{a}}_{A_n}, T^{\bar{a}}, \delta^{\bar{a}}, U^{\bar{a}}, RP^a_{A_1} \dots RP^a_{A_n} \}$$

where for agent a, $RP^a_{A_i}$ is the reserve price for issue A_i, T^a is the deadline for completing the negotiation for all issues, δ^a is the discount factor and U^a is the utility function. For agent $\bar{a}$, theses parameters are defined analogously.

The utility function for agent a is defined as:

$$U^a\left(x^a_{A_1 t} \dots x^a_{A_n t}, t\right) = \sum_{i=1}^{i=n} \left(RP^a_{A_i} - x^a_{A_i t}\right)(\delta)^t$$

where $x^a_{A_i t}$ is the offer made by agent a at time t for issue t. For agent $\bar{a}$, the utility function can be defined similarly.

The negotiation proceeds as a series of rounds where in each round one agent acts as the proposer and the other as the responder, exchanging the roles after each round. The agent begins by making an offer that gives it the highest utility and then yields utility to its opponent by making offers that give it successively lower utility as negotiation proceeds. There are three possible actions that an agent can take on each received offer: accept, quit or counter-offer. The action that an agent takes depends on its strategy.

For the package deal procedure, agent a's strategy ((A_{t-1}^a) is defined as follows:

$$\begin{cases} Quit\ if\ t-1 > T^a \\ Accept\ if\ offer\ received\ is\ better \\ than\ counter-offer \\ Counter-offer\ z_t^a\ at\ t\ otherwise \end{cases}$$

where z_t^a is a set of points of the form $\left(x_{A_1 t}^a \ldots x_{A_n t}^a\right)$ that satisfy the following constraint:

$$U^a\left(x_{A_1 t}^a \ldots x_{A_n t}^a, t\right) = U_{max}^a - y_t^a \times U_{max}^a$$

where

$$U_{max}^a = \sum_{i=1}^{i=n} U_{A_i max}^a = \sum_{i=1}^{i=n} \left(RP_{A_i}^a\right) - \sum_{i=1}^{i=n} \left(RP_{A_i}^{\bar{a}}\right)$$

and y_t^a is the yield factor for agent a . In the above equation $U_{A_i max}^a$ is the maximum agent utility for issue A_i for agent a. For agent $\bar{a}$, $U_{max}^{\bar{a}}$ can be defined similarly.

For the parallel issue-by-issue procedure, agent a's strategy (A_{t-1}^a) is defined as follows:

$$\begin{cases} Quit\ if\ t-1 > T^a \\ Accept\ issue\ A_i\ if\ offer\ received\ is\ better \\ than\ counter-offer \\ Counter-offer\ x_{A_i t}^a\ if\ A_i\ is\ not\ agreed \end{cases}$$

Strategies of agent $\bar{a}$ for package-deal and parallel issue-by-issue negotiation procedures can be defined analogously.

3 Optimal Agenda and Negotiation Procedure

An agenda A_0^a is agent a's optimal agenda if, for *either issue-by-issue* or *package deal* procedure, it gives the agent the maximum utility between all the possible agendas [2]. An agent's utility from an agenda depends on the set of possible outcomes that both the agents prefer over no deal. This set of outcomes is the zone of agreement for the agenda. An *n-issue* negotiation model will have 2^n scenarios based on the agents' mutual preference for each issue. Of these, $n+1$ scenarios are distinct based on the number of issues having a zone of agreement.

The zone of agreement can be explained geometrically by plotting a graph between the utilities of the two agents for each issue. The plot for each issue is called its utility frontier. If the utility frontier lies in the first quadrant, it implies a positive utility for both the agents and thus 3 they have a zone of agreement. If the utility frontier lies in the second or fourth quadrant, only one agent will get a positive utility. A utility frontier in the third quadrant implies that both the agents prefer no deal for that issue.

3.1 Single Issue Negotiation

Let A_1 be the issue negotiated between agents a and $\bar{a}$. This issue will appear in agent a's optimal agenda if this issue will give the agent a positive utility i.e. a prefers an agreement over no deal. Thus $A_0^a(A_1) =$

$$\begin{cases} \{A_1\}, & if\ A_1\ gives\ the\ agent\ a\ positive\ utility \\ \emptyset, & otherwise \end{cases}$$

If utility frontier for issue A_1 lies in the first or fourth quadrants, agent a prefers an agreement on this issue over no deal. A utility frontier in the first or second quadrants implies that the agent $\bar{a}$ prefers an agreement on this issue over no deal. If the utility frontier lies in the third quadrant, both the agents have $\emptyset$ as their optimal agenda.

As there is only a single issue to negotiate, the concept of package deal makes no distinction from that of issue-by-issue.

3.2 Two Issue Negotiation

Let A_1 and A_2 be the two issues under negotiation. From [2], we have three scenarios as follows:

3.2.1 Scenario S_0

In this scenario, both issues have a zone of agreement. Thus both agents prefer an agreement on issues A_1 and A_2 over no deal. So the cumulative utility arising from both the issues will be greater than that of each issue. So the optimal agenda will contain both issues for both the agents.

$$A_0^a(A_1, A_2) = A_0^{\bar{a}}(A_1, A_2) = \{A_1, A_2\}$$

Let $\Delta U_{A_1}^a$ denote the increase in agent a's utility due to a unit change in its allocation for issue A_1 and C_{A_1} denote $\Delta U_{A_1}^a / \Delta U_{A_1}^{\bar{a}}$. C_{A_1} is called the agents' comparative interest in issue A_1. If $C_{A_1} \neq C_{A_2}$, each agents' utility from the package deal is no worse than its utility from parallel issue-by-issue negotiation. If $C_{A_1} = C_{A_2}$, package deal and parallel issue-by-issue negotiation generate the same outcome.

3.2.2 Scenario S_1

In this scenario, only one issue has a zone of agreement (assume it to be issue A_1). Thus both agents prefer an agreement on issue A_1 over no deal. For issue A_2, agents can have three possible preferences. First, a may prefer an agreement on issue A_2 to no deal, while $\bar{a}$ prefers no deal to an agreement on issue A_2. Second, $\bar{a}$ may prefer an agreement on issue A_2 to no deal while a prefers no deal to an agreement. Finally, neither agent may prefer an agreement on issue A_2 to no deal.

$$A_0^a(A_1, A_2) = \{A_1\} \cup A_0^a(A_2)$$

$$A_0^{\bar{a}}(A_1, A_2) = \{A_1\} \cup A_0^{\bar{a}}(A_2)$$

The optimal procedure for agenda $\{A_1, A_2\}$ is the package deal if $C_{A_1} \neq C_{A_2}$ and if $C_{A_1} = C_{A_2}$, both procedures generate the same outcomes.

3.2.3 Scenario S_2

In this scenario, neither issue has a zone of agreement. Since, neither agent prefers an agreement on the two issues to no deal, each of the two issues if discussed separately would lead to conflict. However, if both issues are negotiated together, the utilities for each issue get added up and may result in a zone of agreement. In that case, both the issues will form the optimal agenda. Thus the optimal procedure is package deal. The optimal agenda is given by

$$A_0^a(A_1, A_2) = A_0^a(A_1) \cup A_0^a(A_2)$$

$$A_0^{\bar{a}}(A_1, A_2) = A_0^{\bar{a}}(A_1) \cup A_0^{\bar{a}}(A_2)$$

3.3 Three Issue Negotiation

Let A_1, A_2 and A_3 be the three issues under negotiation. For three issues we have eight possible scenarios, of which four are distinct.

3.3.1 Scenario S_0

In this scenario, all the three issues have a zone of agreement. Thus both the agents prefer an agreement on issues A_1, A_2 and A_3 over no deal and their utility frontiers will lie in the first quadrant. So the cumulative utility arising from all the issues will be greater than that of each issue. So the optimal agenda will contain all the three issues for both the agents. In this scenario, there is only one optimal agenda for both the agents.

$$A_0^a(A_1, A_2, A_3) = A_0^{\bar{a}}(A_1, A_2, A_3) = \{A_1, A_2, A_3\}$$

If $C_{A_1} = C_{A_2} = C_{A_3}$, the utility frontiers for the three issues are parallel lines and the cumulative utility frontier is a single straight line as seen in Fig.1. So there is no difference between parallel *issue-by-issue* and *package deal* and they produce the same result. Let E,F and G be the points on the utility frontiers of issue A_1, A_2 and A_3 at which agreement occurs when each of them are individually negotiated. Let H be the point of agreement for parallel issue-by-issue negotiation of these issues. Since no

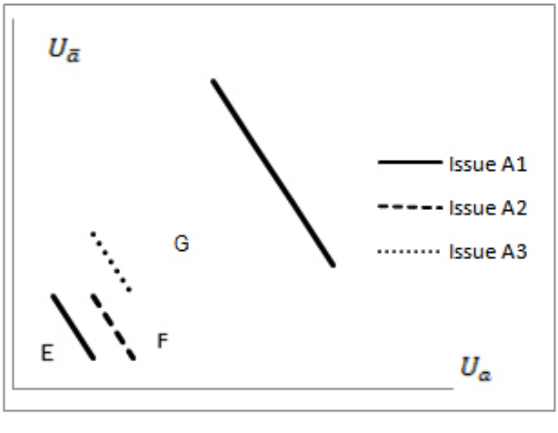


Fig. 1. $C_{A_1} = C_{A_2} = C_{A_3}$

trade-offs are possible as the sum is a straight line, the package deal will also give H as the outcome.

In all other cases: (The Fig. 2 shows the case when all are unequal i.e. $C_{A_1} \neq C_{A_2} \neq C_{A_3}$). Let t_e be the equilibrium time for single issue negotiation. Let E, F and G be the points on the utility frontiers of issue A_1, A_2 and A_3 at which agreement occurs when each of them are individually negotiated. Let H be the point of agreement for parallel *issue-by-issue* negotiation of these issues. For *package deal* at t_e agent $\bar{a}$ will offer a set of triples from line RS and agent a from the line PQ. For all $t' < t_e$ both the agents will make offers that give them higher utilities than the equilibrium value. Let R′P′ and P′Q′ be the offers that $\bar{a}$ and a make respectively at such a t′. Then, they will reach an agreement at P′. Suppose at time $t_e - 1$ agent $\bar{a}$ makes the offer that gives it a higher utility than the value at P and the agent a makes an offer giving it a higher utility than S, then at t_e they will reach an agreement at either P or S depending on whether her it is $\bar{a}$'s or a's turn to offer. Thus for package deal, the agreement lies in the segment PP′S all of which dominates over H. Thus package deal produces a result no worse than parallel issue-by-issue. Hence, the optimal procedure is package deal.

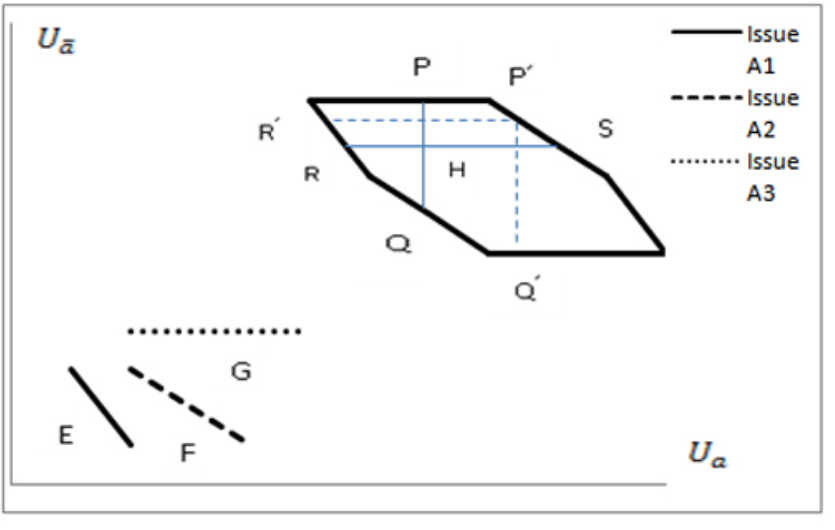


Fig. 2. $C_{A_1} \neq C_{A_2} \neq C_{A_3}$

3.3.2 Scenario S_1

In scenario S_1, two issues have a zone of agreement (assuming them to be issues A_1 and A_2). Thus both the agents prefer an agreement on issue A_1 and A_2 to no deal. For issue A_3, the agents can have three possible preferences. First, a may prefer an agreement on issue A_3 to no deal, while $\bar{a}$ prefers no deal to an agreement on issue. Second, $\bar{a}$ may prefer an agreement on issue A_3 to no deal while a prefers no deal to an agreement. Finally, neither agent may prefer an agreement on issue A_3 to no deal. In this scenario, there are three possible optimal agendas for the two agents:

$$A_0^a(A_1, A_2, A_3) = \{A_1, A_2\} \cup A_0^a(A_3)$$

$$A_0^{\bar{a}}(A_1, A_2, A_3) = \{A_1, A_2\} \cup A_0^{\bar{a}}(A_3)$$

The optimal procedure for agenda $\{A_1, A_2, A_3\}$ is as given in scenario S_0 discussed in section3.3.1. The optimal procedure for agenda $\{A_1, A_2\}$ is *package deal* if $C_{A_1} \neq C_{A_2}$ and parallel *issue-by-issue* if $C_{A_1} = C_{A_2}$.

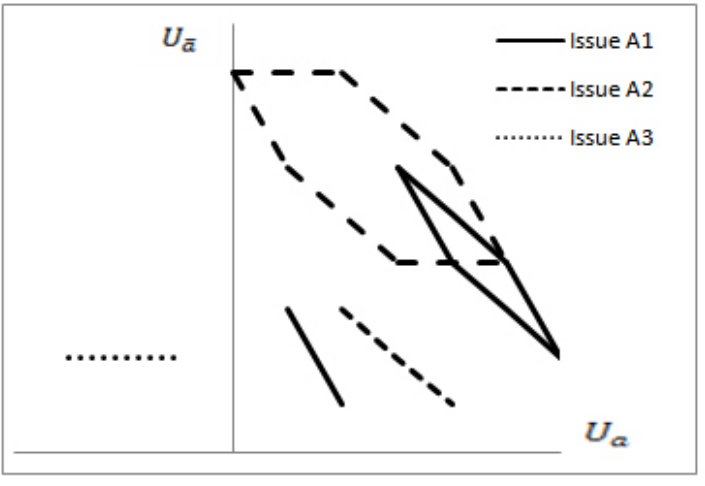


Fig. 3. Scenario S_1

In Fig. 3, the solid graph gives the cumulative utility for issues A_1 and A_2. The broken graph corresponds to that of all three issues. As can be seen, adding the utility of issue A_3 lowers the cumulative utility of agent a and increases that of the agent $\bar{a}$. Hence the optimal agendas will be

$$A_0^a(A_1, A_2, A_3) = \{A_1, A_2\},\ A_0^{\bar{a}}(A_1, A_2, A_3) = \{A_1, A_2, A_3\}$$

3.3.3 Scenario S_2

In this scenario, only one issue has a zone of agreement (assuming it to be A_1). Thus both the agents prefer an agreement on issue A_1 to no deal. The other two issues can lie in the second, third or fourth quadrants depending on the preferences of the two agents. In this scenario, the optimal agenda is given by:

$$A_0^a(A_1, A_2, A_3) = \{A_1\} \cup A_0^a(A_2, A_3)$$

$$A_0^{\bar{a}}(A_1, A_2, A_3) = \{A_1\} \cup A_0^{\bar{a}}(A_2, A_3)$$

The optimal procedure will be as in Scenario S_1 discussed in section 3.3.2.

3.3.4 Scenario S_3

In this scenario, no issue has a zone of agreement. Thus both the agents prefer no deal to an agreement on all issues and individual negotiations on the issues will certainly

lead to a conflict. However, it is possible that when more than one issue is taken together it may result in a zone of agreement. Consequently, the optimal agendas in this scenario can be as follows:

$$A_0^a(A_1, A_2, A_3) = \{A1, A2, A3\}\ or\ \{A1, A2\}\ or\ \{A2, A3\}\ or\ \{A1, A3\}\ or\ \emptyset$$

The optimal procedure will *be package deal* as *issue-by-issue* will always lead to conflict.

In Fig. 4, although no issue has a zone of agreement, adding the utilities of issues A_1 and A_2 leads to a zone of agreement. This is represented by that part of the graph which is in the first quadrant. Issue A_3 will not figure in the optimal agenda for either agents since this will decrease the cumulative utilities of both the agents.

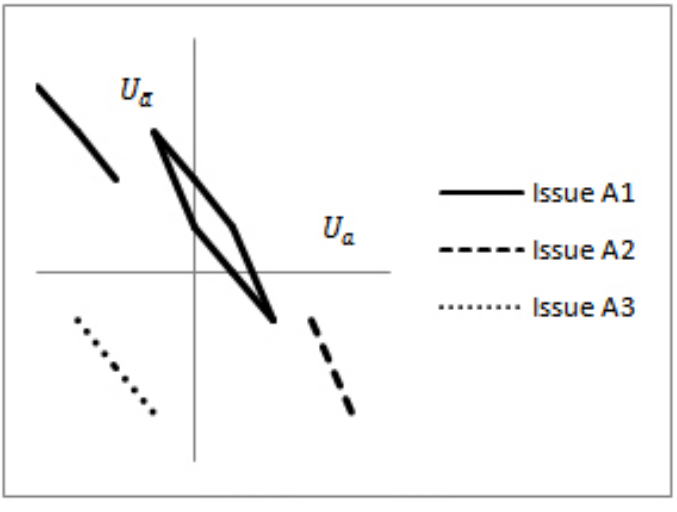


Fig. 4. Scenario S_3

3.4 N- Issue Negotiation

As stated earlier, when *n-issue*s are negotiated there will be a total of 2^n scenarios. Elimination of the symmetric cases leaves us with $n + 1$ distinct scenarios. Let S_i denotes the scenario where i issues do not have a zone of agreement. The remaining $n - i$ (assuming them to be $A_1 \cdots A_{n-i}$) issues will have a zone of agreement and their utility frontiers will lie in the first quadrant. For these $n - i$ issues, both the agents prefer a settlement and thus the optimal agenda for both the agents in scenario S_i will include these issues.

The remaining i issues can fall into the following categories:

- Agent $\bar{a}$ prefers an agreement while the agent a does not.
- Agent $\bar{a}$ does not prefer while agent a prefers an agreement.
- Both the agents do not prefer an agreement.

Accordingly, their utility frontiers will lie in the second, third and fourth quadrant respectively. Those issues in the third quadrant need not be considered as adding them will reduce the overall utility for both the agents.

For scenario $S_i(i \neq 0, n)$, the optimal agenda for agents a and $\bar{a}$ are respectively:

$$A_0^a(A_1 \ldots A_n) = \{A_1 \ldots A_{n-i}\} \cup A_0^a(A_{n-i+1} \ldots A_n)$$

$$A_0^{\bar{a}}(A_1 \ldots A_n) = \{A_1 \ldots A_{n-i}\} \cup A_0^{\bar{a}}(A_{n-i+1} \ldots A_n)$$

For scenario S_0, $A_0^a(A_1 \ldots A_n) = A_0^{\bar{a}}(A_1 \ldots A_n) = \{A_1 \ldots A_n\}$

For scenario S_n, $A_0^a(A_1 \ldots A_n) = \bigcup_{i=1}^{i=n} A_0^a(A_i)$; and analogously for $A_0^{\bar{a}}$.

If $C_{A_1} = C_{A_2} = \cdots = C_{A_n}$, the *package deal* and *parallel issue-by-issue* produce the same outcome. In this case, the cumulative utility frontier is a single straight line and so no trade-offs are possible. So there is no difference between the *parallel issue-by-issue* and *package deal* and they produce the same result.

If there exists some i, j such that $C_{A_i} \neq C_{A_j}, i \neq j, 0 \leq i, j \leq n$, then the package deal produces a result no worse *parallel issue-by-issue*.

Proof: This can be proved by the principle of mathematical induction for Scenario S_0. The result is true for two-issue model as has been proved in section 3.2. Assume the result is true for all $n \leq k$.

Case 1: $C_{A_1} = C_{A_2} = \cdots = C_{A_k}$

In this case the cumulative utility of all the issues is a single straight line. Now, this can be considered as the utility frontier for a single issue. If $C_{A_{k+1}} = C_{A_k}$, the resulting cumulative utility will again be a straight line as given by Scenario S_0 of the two-issue model, both the *package deal* and *parallel issue-by-issue* producing the same outcome. If $C_{A_{k+1}} = C_{A_k}$, the resulting cumulative utility will be a parallelogram as was seen in the two issue model when $C_{A_1} \neq C_{A_2}$. Thus it follows that *package deal* produces a result that is no worse than the parallel issue-by-issue negotiation procedure.

Case 2: $C_{A_i} \neq C_{A_j}$, $for\ some\ i \neq j, 0 \leq i, j \leq k$

If $C_{A_{k+1}} = C_{A_i}\ for\ some\ i \leq k$ or $C_{A_i} = C_{A_j}\ for\ some\ i, j \leq k+1$ the cumulative utility frontier of these two issues can be considered as a single issue. Thus this case will correspond to a negotiation involving at most k issues. So, package deal produces a result that is no worse than *parallel issue-by-issue* as per the inductive hypothesis.

If $C_{A_{k+1}} \neq C_{A_i}\ for\ all\ i \leq k$ and $C_{A_i} \neq C_{A_j}\ for\ all\ i, j \leq k$ then the comparative utilities of all the issues are different. Let $U = \{A_1, A_2, \ldots, A_{k+1}\}$. Form $k+1$ distinct sets $S_1, S_2, \ldots, S_{k+1}$ such that $for\ all\ i, S_i \subset U\ and\ |S_i| = k$. Let X_i and Y_i represent the cumulative utility of an agent obtained by package deal and parallel issue-by-issue negotiation for set S_i respectively. Now, $X_i \geq Y_i, for\ all\ i$ by the inductive hypothesis. Summing up all Y_i will give a utility that is k times the equilibrium utilities of each of the $k+1$ issue when negotiated separately, i.e. this sum represents k times the cumulative utility obtained by the parallel issue-by-issue negotiation of the $k+1$ issues. Sum of all X_i represents the cumulative utility of $k+1$ different sets of k distinct issues each. This sum is equal to the cumulative utility of k different sets (let utility of each be Z_i) of $k+1$ distinct issues each formed from the same $k.(k+1)$ issues. Now, since there are only $k+1$ distinct issues among these $k.(k+1)$ issues, each of these k sets will contain the same $k+1$ issues and thus will have the same cumulative utility i.e. the cumulative utility obtained by the package deal negotiation procedure of the $k+1$ issues.

$$k.U_{PD} = \sum_{i=1}^{k} Z_i = \sum_{i=1}^{k+1} X_{i\ i} \geq \sum_{i=1}^{k+1} Y_{i\ i} = k.U_{IBI}$$

i.e. $U_{PD} \geq U_{IBI}$ where U_{PD} and U_{IBI} represent the utilities obtained from the *package deal* and *parallel issue-by-issue* negotiation procedures respectively.

For all the other scenarios, if the optimal agenda has $k+1$ issues, the proof becomes the same as that of Scenario S_1. If the optimal agenda contains only i issues $i < k+1$, then again by the inductive hypothesis the result becomes true. Thus, we have the result that the package deal produces a result that is no worse than the parallel issue-by-issue negotiation. Thus by induction we prove the above result for n-issue negotiation

4 Conclusion

The effect of varying the agenda and procedure on the negotiation outcome were studied. Initially, an equilibrium strategy was defined for both the agents for both the *package deal and parallel issue-by-issue* negotiation procedures. The different scenarios under which negotiation can take place were then determined. On the basis of this strategy, the optimal agenda-procedure combinations for the different scenarios were determined. This was first done for a 3-issue model and later extended to an n-issue negotiation model. It is found that different agenda-procedure combinations will have a marked impact on the negotiation outcome and it is possible for the agents to have identical preferences for the agenda-procedure combination.

References

[1] Fatima, S.S., Wooldridge, M., Jennings, N.R.: Optimal Agendas for Multi IssueNegotiation. In: AAMAS 2003, Melbourne Australia, July. ACM, New York (2003)

[2] Fatima, S.S., Wooldridge, M., Jennings, N.R.: Optimal Negotiation of Multiple Issues in Incomplete Information settings. In: AAMAS 2004, July. ACM, New York (2004)

[3] Fershtman, C.: A Note on Multi-issue two-sided Bargaining: Bilateral Procedures. Games and Economic Behaviour (2000)

[4] Lai, G., Sycara, K., Li, C.: A Decentralized Model for Multi Attribute Negotiations. In: Proceedings of ICEC 2006 August Fredericton, Canada. ACM, New York (2006)

[5] Jennings, N.R., Faratin, P., Loumuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated Negotiation: Prospects, Methods and Challenges. Journal of Group Decision and Negotiation 10(2) (2001)

[6] Garcia, A., Giese, H., Romanovsky, A., Choren, R., Leung, H.-f., Lucena, C., Klein, F., Platon, E.: Software Engineering for Large Scale Multi Agent Systems-SELMAS 2006: Workshop Report, ACM SIGSOFT Software Engineering Notes, September 2006, vol. 31(5), pp. 24–31 (2006)

Toward an AgentSpeak(L) Theory of Commitment and Intentional Learning

Alejandro Guerra-Hernández[1], José Martín Castro-Manzano[1], and Amal El-Fallah-Seghrouchni[2]

[1] Departamento de Inteligencia Artificial
Universidad Veracruzana
Facultad de Física e Inteligencia Artificial
Sebastián Camacho No. 5, Xalapa, Ver., México, 91000
aguerra@uv.mx, e_f_s_s@hotmail.com
[2] Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie
Avenue du Président Kennedy, Paris, France, 75016
Amal.Elfalah@lip6.fr

Abstract. This work is about the commitment strategies used by rational agents programmed in *AgentSpeak*(L) and the relationship between single-minded commitment and intentional learning. Although agent oriented languages were proposed to reduce the gap between theory and practice of Multi-Agent Systems, it has been difficult to prove BDI properties of the agents programmed in such languages. For this reason, we introduce some ideas to reason temporally about the intentional state of rational agents in order to prove what kind of commitment strategy is used by *AgentSpeak*(L) agents, based on the operational semantics of the programming language. This enables us to prove that any agent programmed in this language follows by default a limited form of single-minded commitment. Then we analyze how intentional learning can enhance this commitment strategy allowing preemptive abandon of intentions.

1 Introduction

The philosophical foundation for intentional agency is provided by the theory of practical reasoning proposed by Bratman [3]. This theory is innovative because it does not reduce intentions to some combination of beliefs and desires, but indeed it assumes that they are composed by hierarchical, partial plans. Such assumption explains better temporal aspects of practical reasoning as future intentions, persistence, reconsideration and commitment. Three kinds of reconsideration (and nonreconsideration) are identified by Bratman: nondeliberative, based on habits; deliberative, based on belief-desire reasons; and policy based. Since reconsidering the intention that α always open the question of whether α, reconsideration is closely related to commitment. We are interested on how would policy based commitment strategies be approached by intentional learning.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 848–858, 2008.

Different multi-modal BDI (Belief-Desire-Intention) logics [11,13,14] formalise practical reasoning. They are used to reason about rational agents because of their expressiveness, but not to program them. Agent oriented languages, as *AgentSpeak*(L) [10], were proposed to reduce the gap between the theory and practice of Multi-Agent Systems (MAS). They have a well defined operational semantics, but verifying intentional properties of the agents programmed in them is not evident, since they dropped intentional and time modalities for the sake of efficiency.

The main question approached in this work is what kind of commitment is used by the rational agents implemented in *AgentSpeak*(L)? In order to answer that, we develop some ideas to reason temporally about intentional operators based on the operational semantics of this language. Then we prove that these agents follow a limited form of single minded commitment [9]. Finally, we discuss the use of intentional learning [6,7,8] to approach full single minded commitment in a similar way, Bratman argues, we form policies of reconsideration.

The paper is organized as follows: section 2 introduces briefly the subject of the commitment strategies in the BDI logics. Section 3 presents the agent oriented language *AgentSpeak*(L) and its operational semantics. Section 4 explains the methodology used to prove BDI properties in such language. Section 5 presents the results on the temporal reasoning approach we use to inquire about the commitment strategy followed by *AgentSpeak*(L) agents and the strategy found. The role of intentional learning in commitment is introduced briefly in section 6. Section 7 closes with discussion and future work.

2 Commitment

Different computational theories of practical reasoning have been proposed to capture the main ideas proposed by Bratman. Cohen and Levesque [4] defined intentions as choice with commitment, based on the concept of persistent goals. A critical examination of this theory [12] suggested that it fails to capture important aspects of commitment, as no-infinite deferral. Alternatively, commitment has been approached as a process of maintenance and revision of intentions, relating current and future intentions. Different types of commitment strategies define different types of agents. Three of them have been extensively studied in the context of BDI_{CTL} [11], where CTL [5] is the well known branched temporal logic:

- **Blind commitment.** An agent intending that inevitably (A, for all time branches) eventually ($\Diamond$, in a branch) is the case that ϕ, inevitably maintains his intentions until (U) he actually believes ϕ (his intention is achieved):

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \mathsf{BEL}(\phi)) \tag{1}$$

- **Single-minded commitment.** An agent maintains his intentions as long as he believes they are not achieved or optionally (E) eventually are achievable:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)) \tag{2}$$

– **Open-minded commitment.** An agent maintains his intentions as long as they are not achieved or they are still desired:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{DES}(\mathsf{A}\Diamond\phi))) \quad (3)$$

BDI_{CTL} is expressive enough to capture these notions of commitment. It is possible to verify if an agent system satisfies them, proving they are valid formulae in such system. Convergence to what is intended, under certain conditions, can also be proved [9]. The problem is that the multimodal logics of rational agency, as BDI_{CTL}, were conceived to reason about agents and not to program them. Agent oriented languages, as $AgentSpeak(L)$, were proposed to fill the gap between the theory and practice of MAS.

3 AgentSpeak(L)

The grammar of $AgentSpeak(L)$ [10], as defined for its interpreter Jason [2], is shown in table 1. As usual an agent ag is formed by a set of plans ps and beliefs bs. Each belief $b_i \in bs$ is a ground first-order term. Each plan $p \in ps$ has the form $trigger\ event : context \leftarrow body$. A trigger event can be any update (addition or deletion) of beliefs (at) or goals (g).The context of a plan is an atom, a negation of an atom or a conjunction of them. A non empty plan body is a sequence of actions (a), goals, or belief updates. $\top$ denotes empty elements, e.g., plan bodies, contexts, intentions. Atoms (at) can be labelled with sources. Two kinds of goals are defined, achieve goals (!) and test goals (?).

Table 1. Grammar of $AgentSpeak(L)$ [2]

$ag ::= bs \quad ps$	$at ::= P(t_1, \ldots, t_n) \quad (n \geq 0)$
$bs ::= b_1 \ldots b_n \quad (n \geq 0)$	$\mid P(t_1, \ldots, t_n)[s_1, \ldots, s_m] \quad (n \geq 0, m \geq 0)$
$ps ::= p_1 \ldots p_n \quad (n \geq 1)$	$s ::= \texttt{percept} \mid \texttt{self} \mid id$
$p ::= te : ct \leftarrow h$	$a ::= A(t_1, \ldots, t_n) \quad (n \geq 0)$
$te ::= +at \mid -at \mid +g \mid -g$	$g ::= !at \mid ?at$
$ct ::= ct_1 \mid \top$	$u ::= +b \mid -b$
$ct_1 ::= at \mid \neg at \mid ct_1 \wedge ct_1$	
$h ::= h_1; \top \mid \top$	
$h_1 ::= a \mid g \mid u \mid h_1; h_1$	

The operational semantics [2] of the language, is given by a set of rules that define a transition system between configurations $\langle ag, C, M, T, s\rangle$, where:

– ag is an agent program formed by a set of beliefs bs and plans ps.
– An agent circumstance C is a tuple $\langle I, E, A\rangle$, where: I is a set of intentions $\{i, i', \ldots\}$, each $i \in I$ is a stack of partially instantiated plans $p \in ps$; E is a set of events $\{(te, i), (te', i'), \ldots\}$, each te is a trigger event and each i is an intention (internal events) or the empty intention $\top$ (external events); and A is a set of actions to be performed in the environment.

- M is a tuple $\langle In, Out, SI \rangle$ working as a mailbox, where: In is the mailbox of the agent; Out is a list of messages to be delivered by the agent; SI is a register of suspended intentions (intentions that wait for an answer message).
- T is a tuple $\langle R, Ap, \iota, \epsilon, \rho \rangle$ that registers temporary information as follows: R is the set of relevant plans for a given event; Ap is the set of applicable plans (the subset of applicable plans which contexts are believed true); ι, ϵ, and ρ register the current intention, event and applicable plan along one cycle of execution.
- The label s indicates the current step in the reasoning cycle of the agent.

Figure 1 shows the interpreter for $AgentSpeak(L)$ as a transition system. The operational semantics rules [2] define the transitions. Because of space limitations, table 2 shows only the rules that are relevant for the next section.

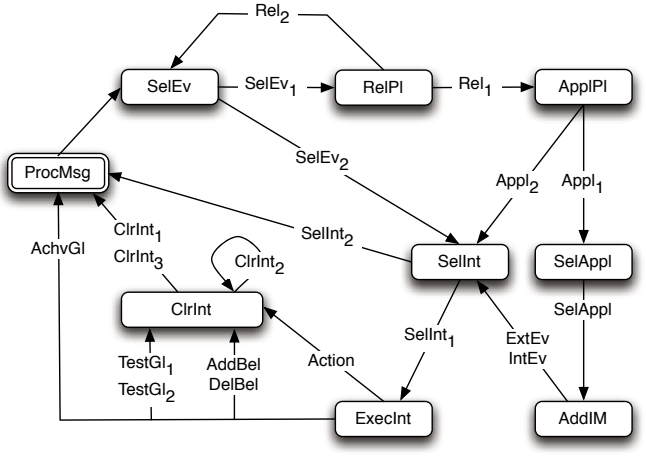


Fig. 1. The interpreter for $AgentSpeak(L)$ as a transition system

Although the operational semantics defines clearly the practical reasoning performed by an agent, it is difficult to prove intentional properties using it. This is due to the abandonment of intentional and temporal modalities. The approach followed in this paper is to define these temporal and intentional operators in terms of the operational semantics, enabling the demonstration of BDI properties.

4 Methodology

Following Bordini [1] we define the intentional modalities of BDI_{CTL} in terms of $AgentSpeak(L)$ operational semantics. First the auxiliary function achievement goals is defined:

$$agoals(\top) = \{\},$$
$$agoals(i[p]) = \begin{cases} \{at\} \cup agoals(i) & \text{if } p = +!at : ct \leftarrow h, \\ agoals(i) & \text{otherwise} \end{cases}$$

which returns the set of atomic formulae (at) subject to an addition of achievement goal ($+!$) in the trigger events of the plans composing a given intention ($i[p]$ denotes an intention i which top plan is p).

Then the operators for BEL, DES, and INTEND are defined in terms of an agent ag and its circumstance C, given in a configuration:

$$\mathsf{BEL}_{\langle ag,C\rangle}(\phi) \equiv bs \models \phi. \tag{4}$$

Table 2. Some rules of the operational semantics of $AgentSpeak(L)$, relevant for the definition of intentional and temporal operators

(SelEv$_1$)	$\frac{\mathcal{S}_{\mathcal{E}}(C_E)=\langle te,i\rangle}{\langle ag,C,M,T,SelEv\rangle \longrightarrow \langle ag,C',M,T',RelPl\rangle}$	s.t. $C'_E = C_E \setminus \{\langle te,i\rangle\}$, $T'_\epsilon = \langle te,i\rangle$
(Rel$_1$)	$\frac{T_\epsilon=\langle te,i\rangle, RelPlans(ag_{ps},te)\neq\{\}}{\langle ag,C,M,T,RelPl\rangle \longrightarrow \langle ag,C,M,T',AppPl\rangle}$	s.t. $T'_R = RelPlans(ag_{ps},te)$
(Rel$_2$)	$\frac{RelPlans(ps,te)=\{\}}{\langle ag,C,M,T,RelPl\rangle \longrightarrow \langle ag,C,M,T,SelEv\rangle}$	
(Appl$_1$)	$\frac{ApplPlans(ag_{bs},T_R)\neq\{\}}{\langle ag,C,M,T,ApplPl\rangle \longrightarrow \langle ag,C,M,T',SelAppl\rangle}$	s.t. $T'_{Ap} = AppPlans(ag_{bs},T_R)$
(SelAppl)	$\frac{\mathcal{S}_{\mathcal{O}}(T_{Ap})=(p,\theta)}{\langle ag,C,M,T,SelAppl\rangle \longrightarrow \langle ag,C,M,T',AddIM\rangle}$	s.t. $T'_\rho = (p,\theta)$
(ExtEv)	$\frac{T_\epsilon=\langle te,\top\rangle, T_\rho=(p,\theta)}{\langle ag,C,M,T,AddIM\rangle \longrightarrow \langle ag,C',M,T,SelInt\rangle}$	s.t. $C'_I = C_I \cup \{[p\theta]\}$
(SelInt$_1$)	$\frac{C_I\neq\{\},\mathcal{S}_{\mathcal{I}}(C_I)=i}{\langle ag,C,M,T,SelInt\rangle \longrightarrow \langle ag,C,M,T',ExecInt\rangle}$	s.t. $T'_\iota = i$
(SelInt$_2$)	$\frac{C_I=\{\ \}}{\langle ag,C,M,T,SelInt\rangle \longrightarrow \langle ag,C,M,T,ProcMsg\rangle}$	
(AchvGl)	$\frac{T_\iota=i[head\leftarrow !at;h]}{\langle ag,C,M,T,ExecInt\rangle \longrightarrow \langle ag,C',M,T,ProcMsg\rangle}$	s.t. $C'_E = C_E \cup \{\langle +!at,T_\iota\rangle\}$, $C'_I = C_I \setminus \{T_\iota\}$
(ClrInt$_1$)	$\frac{T_\iota=[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C',M,T,ProcMsg\rangle}$	s.t. $C'_I = C_I \setminus \{T_\iota\}$
(ClrInt$_2$)	$\frac{T_\iota=i[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C',M,T,ClrInt\rangle}$	s.t. $C'_I = (C_I \setminus \{T_\iota\}) \cup \{k[(head' \leftarrow h)\theta]\}$ if $i = k[head' \leftarrow g;h]$ and $g\theta = TrEv(head)$
(ClrInt$_3$)	$\frac{T_\iota\neq[head\leftarrow\top]\wedge T_\iota\neq i[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C,M,T,ProcMsg\rangle}$	

$$\mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \equiv \phi \in \bigcup_{i \in C_I} agoals(i) \vee \phi \in \bigcup_{\langle te,i\rangle \in C_E} agoals(i). \quad (5)$$

$$\mathsf{DES}_{\langle ag,C\rangle}(\phi) \equiv \langle +!\phi, i\rangle \in C_E \vee \mathsf{INTEND}_{\langle ag,C\rangle}(\phi). \quad (6)$$

These definitions are used to prove the assymetry thesis [3] (See table 3). The thesis expresses that intention-belief inconsistency is closer to irrationality than intention-belief incompletness (AT1-AT3), and the same for intention-desire (AT4-AT6) and desire-belief (AT7-AT8). Bordini and Moreira [1] prove that, under close world assumption, all *AgentSpeak*(L) agents do not satisfy the asymmetry thesis AT1, AT5, and AT7, but they satisfy the rest of them. This means that *AgentSpeak*(L) is not equivalent to any of the BDI modal systems studied previously by Rao and Georgeff [11].

Table 3. Asymmetry thesis expressed in *AgentSpeak*(L)

Label	Theorem
AT1	$\models \mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \implies \mathsf{BEL}_{\langle ag,C\rangle}(\phi)$
AT2	$\not\models \mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \implies \mathsf{BEL}_{\langle ag,C\rangle}(\phi)$
AT3	$\not\models \mathsf{BEL}_{\langle ag,C\rangle}(\phi) \implies \mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$
AT4	$\models \mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \implies \mathsf{DES}_{\langle ag,C\rangle}(\phi)$
AT5	$\not\models \mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \implies \mathsf{DES}_{\langle ag,C\rangle}(\phi)$
AT6	$\not\models \mathsf{DES}_{\langle ag,C\rangle}(\phi) \implies \mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$
AT7	$\models \mathsf{DES}_{\langle ag,C\rangle}(\phi) \implies \mathsf{BEL}_{\langle ag,C\rangle}(\phi)$
AT8	$\not\models \mathsf{DES}_{\langle ag,C\rangle}(\phi) \implies \mathsf{BEL}_{\langle ag,C\rangle}(\phi)$
AT9	$\not\models \mathsf{BEL}_{\langle ag,C\rangle}(\phi) \implies \mathsf{DES}_{\langle ag,C\rangle}(\phi)$

5 Results

The main contributions of this paper are the preliminary definition of temporal operators to reason about *AgentSpeak*(L) programs; and a demonstration that *AgentSpeak*(L) agents are not blind commited, but perform a limited single-minded commitment.

5.1 Time

Since *AgentSpeak*(L) abandons time modalities for the sake of efficiency, it is necessary to redefine them. As it is well known, temporal modalities are defined after a Kripke Structure $\langle S, R, L\rangle$ where S is a set of states, L the labeling for each state in S and R is a total relation on $S \times S$. Roughly, the states in *AgentSpeak*(L) correspond to agent configurations $\langle ag, C, M, T, s\rangle$; R is defined by the operational semantics of the system, being certainly total ($\forall k\ \exists t\ (k,t) \in R$ s.t. $k, t \in S$), as shown in figure 1. L is the label of primitive formulae valid at a given state. Validity for intentional operators is defined as in the previous section. Paths, as usual, are sequences of states (configurations) $c_0, \ldots, c_n$.

Then, the definition of next is:

$$\models_{c_0} \bigcirc\alpha \;\equiv\; T_\iota = i[head \leftarrow \alpha; h] \tag{7}$$

for $\alpha \in \{a, g, u\}$ (action, goal, or belief update). c_0 is the current configuration, where formulae are evaluated. Observe that $T_\iota = \_$ always, except when an intention has been successfully selected to be executed at time t in $s = \texttt{SelInt}$, so that at time $t + 1$ the system will be in $s = \texttt{ExecInt}$ since the selection was successful (otherwise the system goes to `ProcMsg`), then α will occur in the next state of the system. It is evident that we can now define the semantics for expressions like $\bigcirc\mathsf{BEL}(\phi)$, at least for intentional updates, e.g., changes that result from the execution of intentions.

As part of our current work, we are exploring formal definitions for until:

$$\models_{c_0} \phi \;\mathsf{U}\; \psi \;\equiv\; \exists k > 0 \;\models_{c_k} \psi \wedge \forall 0 < j \leq k \models_{c_j} \phi \tag{8}$$

and eventually:

$$\models_{c_0} \Diamond\phi \;\equiv\; \exists k > 0 \models_{c_k} \phi \tag{9}$$

With these definitions we already can prove some properties about the commitment strategies of *AgentSpeak*(L) agents.

5.2 Commitment Strategies in *AgentSpeak*(L)

The main question at this stage of our research on commitment and intentional learning was what kind of commitment strategy is used by *AgentSpeak*(L) agents? Knowing that is important, because intentional learning seems to be irrelevant for a blindly committed agent, while it can be really useful and explanatory if agents are single or open minded. So our first step was to prove that *AgentSpeak*(L) agents do not satisfy the blind commitment axiom under no-infinite deferral.

Proposition 1. *AgentSpeak*(L) *agents satisfy the no-infinite deferral axiom* $\mathsf{INTEND}(\phi) \Rightarrow \mathsf{A}\Diamond(\neg\mathsf{INTEND}(\phi))$.

Proof. Given the definition for intend (eq. 5), the no-infinite deferral axiom expresses that if a plan p with context $+!\phi$ is adopted to form an intention, this plan is eventually retired from C_I (active intentions) and C_E (suspended intentions). Given the finite nature of the plans and providing that intentions and events are always possible to be selected in *SelInt* and *SelEv* steps, there are different paths satisfying $\mathsf{A}\Diamond\neg\mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$, all of them via $\bigcirc\neg\mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \equiv \bigcirc -!\phi$ (eq. 7) or $s = ClrInt$: i) Succesful execution of a plan: when the plan body becomes empty, the plan is removed from $i[p] \in C_I$ by $ClrInt_2$; ii) Succesful execution of intention: when an intention becomes empty, the full intention is removed from C_I by $ClrInt_1$; iii) Keep going execution of the intention: when the body of the plan in the top of an intention is not empty, the cycle continues and the intention will be eventually selected again by *SelInt* arriving, if everything goes right, to one of the previous situations. If something goes wrong,

a failure mechanism is activated by an event of the form $\langle -!\phi, i[p]\rangle$. Although Bordini et al. [2] only formalizes failures in finding relevant plans (Rel_2) which discards suspended intentions in C_E, other forms of failure detection have been considered in the context of intentional learning [6,8]. By successful or failed execution of the plans every adopted intention is inevitable eventually dropped. □

Proposition 2. *AgentSpeak(L) agents do not satisfy the blind commitment axiom* $\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \mathsf{BEL}(\phi))$.

Proof. Given that the no-infinite-deferral axiom is safisfied by *AgentSpeak(L)* agents, the blind axiom can be reduced to $\mathsf{INTEND}(\mathsf{A}\Diamond(\phi)) \implies (\mathsf{A}\Diamond(\mathsf{BEL}(\phi))$, given the blind commitment axiom (eq. 1) and assuming weak until [9]. In order to prove the proposition we will define an agent that does not satisfy the reduced blind commitment axiom. Consider an agent s.t. $ag = \langle bs, ps\rangle$ where $bs = \{\}$ and $ps = \{+b(t_1) : \top \leftarrow p(t_2).\ \ +!p(t_2) : \top \leftarrow +b(t_3).\}$ at its initial configuration. Suppose that from perception of the environment a belief $b(t_1)$ is added to the $ag_{bs} = \{b(t_1)\}$. An event is generated by this belief update, so that $C_E = \{\langle +b(t_1), \top\rangle\}$. Then following the state transitions defined by the semantic rules $SelEv_1$, Rel_1, $ApplPl_1$, we obtain a configuration where $C_I = \{[+b(t_1) : \top \leftarrow !p(t_2).]\}$ and $C_E = \{\}$, Then proceeding with the reasoning steps $SelAppl$, $ExtEv$, $SelInt_1$, $AchvGl$ we obtain a configuration where $C_E\langle +!p(t_2), +b(t_1) : \top \leftarrow \top\rangle$, $C_I = \{\}$. At this moment, the agent $\mathsf{DES}_{\langle ag,C\rangle}(p(t_2))$ (eq. 6). If we apply then $SelEv_1$, Rel_1, $AppPl_1$, $SelAppl$ then we obtain a configuration where $C_I = \{[+!p(t_2) : \top \leftarrow\ \ +b(t_3).]\}$ and $C_E = \{\}$, where the agent $\mathsf{INTEND}_{\langle ag,C\rangle}(p(t_2))$ (eq. 5). Then proceeding with $IntEv$, $SelInt_1$, $AddBel$ then $C_E = \langle +b(t_3)[self], \top\rangle$, $ag_{bs} = \{b(t_1)\}$ and $C_I = \{[+b(t_1) : \top \leftarrow \top \ddagger +!p(t_2) : \top \leftarrow \top]$ and $bs = \{b(t_1), b(t_3)\}$. The intention about $p(t_2)$ is maintained. Observe that the plan bodies in the intention are empty, so the $ClrInt$ rules will discard the whole intention, so that $\neg\mathsf{INTEND}_{\langle ag,C\rangle}(p(t_2))$ and $\neg\mathsf{BEL}_{\langle ag,C\rangle}(p(t_2))$. $\mathsf{INTEND}(\mathsf{A}\Diamond(\phi)) \implies (\mathsf{A}\Diamond(\mathsf{BEL}(\phi))$ is not satisfied for this agent. □

In fact, our agent does not safisfy the extended blind commitment axiom (eq. 1), since the agent did not keep its intention about $p(t_2)$ until she believed it. This reasoning is similar to the demonstration of intention-belief incompleteness (AT2) for *AgentSpeak(L)* [1].

Proposition 3. *AgentSpeak(L) agents satisfy a limited single-minded commitment* $\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi))$.

Proof. Following the no-infinite-deferral demonstration, in the failure cases the agent will eventually satisfy $\bigcirc\neg\mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$ because Rel_2 which means that for an event $\langle te, i[+!\phi : c \leftarrow h.]\rangle$ there were not relevant plans and the associated intention will be discarded, e.g., there is not a path of configurations where eventually ϕ, so that it is rational to drop $\mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$. □

This is a limited form of single-mind commitment because $\neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)$ is not represented explicitly by the agent. In fact, she only can not continue intending

ϕ because there are no plans to do it and the full intention fails. It is also limited because of intention-belief incompleteness that can be avoided dropping the close world assumption [1]; or using the intentional and temporal definitions for studying the necessary conditions in the operational semantics and definition of the agents to warrant the expected properties of intentions, e.g., equivalence to a KD modal system.

Intentional learning provides a third alternative approach to achieve a full single-minded strategy, enabling the explicit representation of the reasons to abandon and intention in a less reactive basis, i.e., preemptive abandon of intentions.

6 Intentional Learning and Commitment

An agent can learn about his practical reasons. Learning examples are composed by the beliefs that supported the adoption of a plan as an intention. Because of the no-infinite deferral axiom, all examples become eventually labelled as success or failure cases. Then, first-order induction of logical decision trees is used to learn hypothesis about the reasons for successful adoption of intentions, in order to update the context of plans that have failed. If the examples of an agent do not offer enough evidence to learn, the agent can ask other agents sharing the plan for more examples. We have called this intentional learning [6,7,8].

For example, suppose we have an agent situated in the blocks world who perceives the state shown at figure 2, State-a; and desires to achieve the state shown at State-b. So, he forms an intention after the event $\langle +!on(b,c), \top \rangle$ using a relevant applicable plan. Suppose such plan has the form $[p1]\ +!on(X,Y) : \top \leftarrow .take(X);\ .put(X,Y)$. It means our agent is bold (or naive) about stacking objects, he believes X can be stacked on Y in any circumstance using plan $p1$. Now, suppose that after forming an intention with this plan, the agent perceives the State-c. The internal action $.put(X/a, Y/b)$ will fail, the intention will eventually fail, and the goal $on(b,c)$ will not be achieved.

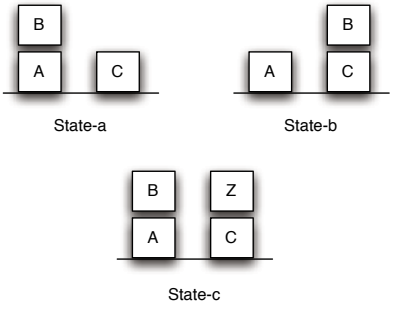


Fig. 2. Blocks world as perceived (State-a), desired (State-b), and perceived after forming an intention to put B on C, but before execute it (State-c)

An intentional learning agent can find out that the right context for the failed plan is $clear(Y) \wedge handfree(Ag)$ and modify his plan definition accordingly. Then the next time he processes the event $\langle +!on(b,c), \top \rangle$ at a state similar to State-c, it will be the case that $bs \not\models clear(c)$, and the plan will not be applicable anymore. This is better than the original non-learning approach, but does not avoid the original problem. Considering also the failure branches of the induced logic tree, in order to add beliefs as $abandon(p1) \leftarrow not(clear(Y))$, enables preemptive abandon of intentions, via a cleaning mechanism to deal events of the form $\langle +abandon(X), i \rangle$, as in Rel_2. We are currently experimenting with different forms of cleaning, implementing this scenario in Jason [2].

7 Discussion and Future Work

We have extended the methodology proposed by Bordini and Moreira [1] to reason about $AgentSpeak(L)$ agents. The extension consists in defining temporal operators based on the operational semantics of this agent oriented programming language. Then we proved that any $AgentSpeak(L)$ agent is not blindly committed, but follows a limited form of single-mind commitment. The main limitations for these agents are intention-belief incompleteness and the lack of an explicit representation for abandoning reasons. We have argued that intentional learning provides a solution for the second problem. Interestingly, modifying the context of the plans in ag_{ps} involves changes in the accessibility while forming intentions. We plan to study if intentional learning can approach intention-belief completeness in this way.

The degree of boldness and cautiousness for a given agent is something hard to define. It is well known that in dynamic environments a very cautious agent performs better than a bold one; and inversely, in static environments boldness pays better. The relevance of learning intentionally is that the right degree of cautioness is learned by the agents, instead of being established once and forever by the programmers.

Immediate future work includes to propose a $CTL_{AgentSpeak(L)}$ logic to reason about these agents. The preliminary results reported here are very encouraging in this sense. The main difficulty here is that CTL is propositional, while the content of $AgentSpeak(L)$ intentional operators is first-order, complicating the definition of L or a valuating function in the Kripke structure supporting temporal semantics. An extended $AgentSpeak(L)$ operational semantics that deals with intentional learning, for both incremental and batch inductive methods, has been proposed [8]. So, it is possible to arrive to a full theory of commitment and intentional learning using the techniques presented here.

This computational theory should consider the concept of policy based reconsideration and commitment in practical reasoning. This is relevant because it brings $AgentSpeak(L)$ closer to its philosophical foundation. But also, because policy based (non)reconsideration seems to be the more interesting of the three cases considered by Bratman. It is not so hard wired as non-deliberative cases, nor is so costly as deliberative ones.

Acknowledgments. The first and third authors are supported by Conacyt CB-2007-1 (project 78910) funding for this research. The second author is supported by Conacyt scholarship 214783.

References

1. Bordini, R.H., Moreira, Á.F.: Proving BDI properties of agent-oriented programming languages. Annals of Mathematics and Artificial Intelligence 42, 197–226 (2004)
2. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley, England (2007)
3. Bratman, M.: Intention, Plans, and Practical Reason. Harvard University Press, Cambridge (1987)
4. Cohen, P., Levesque, H.: Intention is choice with commitment. Artificial Intelligence 42(3), 213–261 (1990)
5. Emerson, A.: Temporal and modal logic. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, pp. 996–1072. Elsevier, Amsterdam (1990)
6. Guerra-Hernández, A., El-Fallah-Seghrouchni, A., Soldano, H.: Learning in BDI Multi-agent Systems. In: Dix, J., Leite, J.A. (eds.) CLIMA 2004. LNCS (LNAI), vol. 3259, pp. 218–233. Springer, Heidelberg (2004)
7. Guerra-Hernández, A., Ortíz-Hernández, G.: Toward BDI sapient agents: Learning intentionally. In: Mayorga, R.V., Perlovsky, L.I. (eds.) Toward Artificial Sapience: Principles and Methods for Wise Systems, pp. 77–91. Springer, London (2008)
8. Guerra-Hernández, A., Ortíz-Hernández, G., Luna-Ramírez, W.A.: Jason smiles: Incremental BDI MAS learning. In: MICAI 2007 Special Session. IEEE, Los Alamitos (in press, 2007)
9. Rao, A.S., Georgeff, M.P.: Modelling Rational Agents within a BDI-Architecture. In: Huhns, M.N., Singh, M.P. (eds.) Readings in Agents, pp. 317–328. Morgan Kaufmann, San Francisco (1998)
10. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: Perram, J., Van de Velde, W. (eds.) MAAMAW 1996. LNCS, vol. 1038, pp. 42–55. Springer, Heidelberg (1996)
11. Rao, A.S., Georgeff, M.P.: Decision procedures for BDI logics. Journal of Logic and Computation 8(3), 293–342 (1998)
12. Singh, M.P.: A critical examination of the Cohen-Levesque Theory of Intentions. In: Proceedings of the European Conference on Artificial Intelligence (1992)
13. Singh, M.P., Rao, A.S., Georgeff, M.P.: Formal Methods in DAI: Logic-Based Representation and Reasoning. In: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pp. 331–376. MIT Press, Cambridge (1999)
14. Wooldridge, M.: Reasoning about Rational Agents. MIT Press, Cambridge (2000)

Collective Pathfinding in Dynamic Environments

Carlos Astengo-Noguez and José Ramón Calzada Gómez

ITESM Campus Monterrey
castengo@itesm.mx, jrcalzada@itesm.mx

Abstract. Pathfinding is a critical element of AI in many modern applications like multiple mobile robots, game industry and flock traffic navigation based on negotiation (FTN). Classical algorithms assume a unique mobile agent with a complete and accurate model of its environment. New applications demand algorithms that consider multiple agents moving in partially known and changing environments. This paper introduces a new algorithm capable of planning paths for multiple agents and takes the FTN scenario as the basis of an example.

1 Introduction

Navigation is the problem of finding a collision-free motion for an agent-system from one configuration (or state) to another [3]. The agent could be a videogame avatar, a mobile robot, or something else. Localization is the problem of using a map to interpret sensor data to determine the configuration of the agent. Mapping is the problem of exploring and sensing an unknown environment to construct a representation that is useful for navigation or localization. Localization and mapping can be combined.

Motion planning has received considerable attention in the research literature (for a good survey see [3] and [8]). There are a number of interesting motion planning tasks such as navigation among moving obstacles, manipulation and grasp planning, assembly planning, and coordination of multiple agents. Most of the work assumes the agent has a complete and accurate model of its environment before it begins to follow its own planned path; less attention has been paid to the problem of multiple (competitive or cooperative) agents moving in such well known environments and even less attention to those who have partially-known environments.

Once the task and the agent system are defined, an algorithm can be chosen based on how it solves the problem. Due to the nature of the problem, it is necessary to find an adequate balance between optimality, completeness, and computational complexity. We must be willing to accept increased computational complexity if we demand optimal motion plans or completeness from our planner.

Based on how the plan is constructed, planners can be divided in two categories. Offline planners construct the plan in advance based on a known model of the environment and pass it on to an executor afterwards. Online planners build the plan incrementally during execution. A similar issue arises in control theory when attempting to distinguish between feed-forward control (commands based

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 859–868, 2008.

on a reference trajectory and dynamic model) and feedback control (commands based on error from the desired trajectory) [3].

Agents are assumed to operate in a planar (R^2) or three dimensional (R^3) environment or (vectorial) space, called workspace W. This workspace will often contain obstacles; let WOi be the i-th obstacle. The free workspace is the set of points

$$W_{free} = W - \bigcup_i WO_i$$

Motion planning, however, does not usually occur in the workspace. Instead, it occurs in the configuration space Q (also called C-space), the set of all agent configurations. Every dot in the configuration space represents a possible arrangement of the agent in the problem space. A contiguous line in configuration space represents a plan, that is, a sequence of planned locations, starting from a starting point (S) and leading to a target (T).

There is a distinction between path planning and motion planning. A path is a continuous curve on the configuration space. It is represented by a continuous function that maps some path parameter, usually taken to be in the unit interval [0, 1], to a curve in Qfree . The choice of unit interval is arbitrary; any parameterization would suffice. The solution to the path planning problem is a continuous function c ∈ C0 such that

c : [0, 1] ? Q where c(0) =qstart, c(1) =qtarget and c(s) ∈ Qfree ∀s ∈ [0, 1].

When the path is parameterized by time t, then c(t) is a trajectory, and velocities and accelerations can be computed by taking the first and second derivatives with respect to time. This means that c should be at least twice-differentiable, i.e., in the class C2. Finding a feasible trajectory is called trajectory planning or motion planning.

There are two main approaches on multiple cooperative pathfinding: Decoupled planning, which plans for each robot independently and coordinates them later; and Centralized planning, which plans the motion of the robots in their "composite" configuration space.

The game industry is demanding algorithms that allow multiple agents to plan for non-colliding routes on congested-dynamical environments [9], such problems also appear in flock traffic navigation (FTN) based on negotiation[2].

FTN uses A* for path planning and, only at intersections, flocks use the Dresner and Stone reservation algorithm [5]. We intend to improve this algorithm using cooperative strategies.

2 Classical (Individual) Pathfinding

The problem in path planning is to find a path for an agent from some location in its environment to another location called target or goal such that obstacles are avoided and a positive cost metric is minimzed. The problem space can be formulated as a set of states connected by directional arcs, each of which has an associated cost. Cost can be defined to be the distance traveled, energy expended, time exposed, etc.

2.1 Pathfinding with A* and D*

A* is a best-first search algorithm that has been widely used in the game industry. It evaluates nodes by combining g(n), the cost to reach the node, and h(n), the estimated cost to get from the node to the goal.

$F(n) = g(n) + h(n)$

A* is complete and optimal if h(n) is an admissible heuristic [10]. D* is an algorithm capable of planning paths in unknown or partially-known dynamic environments in an efficient, optimal and complete manner [12]. The algorithm is formulated as an optimal find-path problem within a directed graph, where the arcs are labeled with cost values that can range over a continuum.

2.2 Cooperative Pathfiding

In cooperative pathfinding each agent is assumed to have full knowledge of all other agents and their planned routes [11]. The complementary problem is called "non-cooperative pathfinding", where the agents have no knowledge of each other's plans and must predict their future movements. There is also another approach called "antagonist pathfinding" where agents try to reach their own goals while preventing other agents from reaching theirs.

There are two possible approaches for multi-agent pathfinding [8],

1. Centralized approach: It takes all agents into account at once and finds all possible solutions. This problem is Pspace-hard.
2. Decoupled or distributed approach: It decomposes the task into independent or weakly-dependent problems for each agent. Each agent can search greedily for a path according to its destination, given the current state of all other agents.

FTN based on negotiation uses a decoupled approach called local repair A*: each agent searches for a route to the destination using A*, ignoring all other agents except for its current neighbors. It is in this neighborhood that negotiation takes place and the flock is created. The agents then follow their route (according to the bone-structure) until a collision is imminent.

It is clear that collisions will happen at the intersections, so there are, with this approach, two possible solutions:

1. Using the Dresner and Stone reservation method [4][5][6][7].
2. Using the Zelinsky [14] brute force algorithm: whenever an agent or a flock is about to move into an occupied position it instead recalculates the remainder of its route.

The implementation of each method depends on the information and the time that the agent or flock has at that particular moment.

The Zelinsky algorithm usually suffers from cycles and other severe breakdowns in scenarios where bottlenecks are present [9] [14]. The Dresner and Stone reservation model was developed for individual agents that can accelerate or decelerate according to the reservation agenda. In simulations performed it was shown that, at the beginning, it works properly, but as time moves on the agents are eventually stopped.

2.3 Cooperative Pathfinding with A*

The task is decoupled into a series of single agent searches. The individual searches are performed in three-dimensional space-time and takes the planned routes of other agents into account. A wait move is included in the agent's action set to enable it to remain stationary. After each agent's route is calculated the states along the route are marked into a reservation table. Entries in the reservation table are considered impassable and are avoided during searches by subsequent agents.

The reservation table represents the agents' shared knowledge about each other's planned routes. The reservation table is a three-dimensional grid: two spatial dimensions and one time dimension. Each Cell that is intersected by the agent's planned route is marked as impassable for precisely the duration of the intersection, thus preventing any other agent from planning a colliding route. Only a small proportion of grid locations will be touched, and so the grid can be efficiently implemented as a hash table [11].

1. New Cooperative Pathfinding Algorithm
Now we can propose a new algorithm that combines D* and Cooperative A*. The algorithms assumes a pre-processing step that calculates preliminary paths for all agents using A*. The A* algorithm must be able to find paths for agents starting from a certain point, moving at a certain speed, and possessing a defined visibility index. The visibility index indicates how far in time can an agent see other agents' paths.

Once the pre-processing step has run, the algorithm loops over the elements of a priority queue. At first agents are added randomly to the queue.

For each agent its path starting from the position occupied at the current time up to the position it will occupy at the current time plus the visibility index is verified against the reservation table. If no collisions are found the agent's delay is set to 0, each step in the path reserved in the table, and its position updated. If a collision was found previous reservation in this step are undone and an altered speed that allows the agent to arrive to the collision point later, in an attempt to avoid it, is calculated.

If the new speed is lower than a pre-determined speed threshold then the agent calculates a new path, in an attempt to minimize its delay. The new path is calculated using A* from the goal to the current position. The algorithm requires the current time, the visibility index and the agent's speed to query the reservation table in order to produce a collision-free path. If the altered speed is not lower than the speed threshold, then the agent's path is adjusted to match the new speed. In either of these scenarios, the delay caused by the collision is calculated.

Finally, if the agent hasn't yet reached its goal, it is added back into the queue, using its delay as the priority, making sure that the most delayed agents have a chance to make reservations in the table before other agents.

```
while !queueIsEmpty() do
i = dequeue()
for t in currentTime...currentTime + visibilityIndex
if reservationTable[path[i][t], t] = -1 then
reservationTable[path[i][t], t] <- i
else
collision <- true
undoReservations(path[i], currentTime, t)
newSpeed <- decreaseSpeed(i, reservationTable[path[i][t], t)
if newSpeed < speedThreshold then
path[i] <- astar(goal[i], path[i][currentTime], reservationTable, currentTime, visibilityIndex, speed)
else
delays[i] <- delays[i] + calculateDelay(path[i], agentSpeeds[i], newSpeed)
agentSpeeds[i] <- newSpeed
adjustPathToSpeed(path[i], agentSpeeds[i])
end
end
end
if collision = false then
agentPositions[i] <- path[i][currentTime]
end
if agentPositions[i] != goal[i] then
enqueue(i, delays[i])
end
end
```

2.4 New Cooperative Pathfinding Algorithm Applied to Flock Traffic Navigation

We will explain the classic FTN based on Negotiation algorithm and then propose an improvement to avoid collisions at intersections.

2.5 Flock Traffic Navigation

Flock Traffic Navigation (FTN) based on negotiation is a new approach for solving traffic congestion problems in big cities [1]. In FTN, vehicles can navigate automatically in groups called flocks allowing 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 issued from multi-agent systems.

The mechanism to negotiate [2] starts with an agent who wants to reach its destination. The agent knows an 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 so-called "bone" structure diagram.

Individual reasoning plays the main role in 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. Decision will be made according to whether they are in Nash equilibrium (There is no incentive for either of them to choose another neighbor agent over the agreed one) or if they are in a Pareto Set [13].

If both agents are in Nash equilibrium, they can travel together as partners and can be benefited with the social bonus. In this moment a new virtual agent is created in order to negotiate with future candidates. Agents in a Pareto Set can be added to this "bone" diagram if their addition benefits them without affecting the original partners negatively. Simulations indicate that flock navigation of autonomous vehicles could substantially save time to users and let traffic flow faster [2].

Until now, Agents make their own path planning according to A* and only at intersections use the Dresner-Stone Algorithm.

2.6 A Collision Detection Flock Traffic Navigation Algorithm

FTN based on Negotiation now are decoupled in two main parts:

```
OFFLINE Algorithm
For each Agent do
plan using A* and find
an optimal path traveling and
an arrival time estimation.

REAL Time Algorithm
For each spirit flock do
A reservation δ-time units forward according to
its path
If reservation = TRUE
Follow previous calculated A*-path
else
Conflict Module
```

The critical issue is the conflict Module that can be changed according to the social rules. Here we present a conflict module according to the rules in [2].

Conflict Module
Rules
Priority 1: Larger Flock goes first
Then tile is marked as obstacle.
Priority 2: If Size of Flocks are equal
compare delay-table
delayed Flock goes first
Then tile is marked as obstacle
Priority 3: If none previous priority is accomplished
Use Stone-Dresner Algorithm

3 Application to Continuous Domains

When applied to computer games, path finding usually takes place on top of one of two representations: A grid structure that wholly describes the traversable game world or a waypoint graph that samples the continuous space over which game agents can move.

The first variant can usually be seen in Real-Time Strategy Games (RTS) and bi-dimensional role-playing games (RPG). The cooperative path finding algorithm in dynamic environments is a perfect fit for these representations and allows game agents to react realistically to the presence of other agents and unforeseen obstacles. Other genres, however, require the simulation of a continuous space updated in fixed time-steps.

Applying a grid-like structure to such spaces can be prohibitively expensive. The algorithm can be modified to work in continuous domains by replacing the reservation table with an analysis of world geometry.

The algorithm consists of an update function, responsible for advancing the state of each agent by a single time-step. The agents are stored in a priority queue that uses each agent's total delays as its key. The function calculates the time elapsed between the last and the current call and uses this value to update the agents. The agents are updated by first performing a collision-detection test between the Minkowski sum of the agent's path and an assigned bounding volume and each of the Minkowski sums of the other agents' predicted paths and their bounding volumes. If a collision is detected, the agent will try to adjust its speed to avoid the intersection point at the intersection time. If this value falls under a specified threshold, the agent will, instead, attempt to calculate a different path.

The predicted paths are calculated using only the other agent's current position, orientation, and speed. These predictions are also limited by the forecasting index, which indicates how far in time are agents willing to predict.

This approach allows the path-planning operation to be distributed across frame updates and the individual update steps for each agent cause no side-effects, making them trivially parallelizable. The algorithm can be further

optimized by pruning the collision-detection search by using spatial partitioning schemes such as kd-Trees and by limiting the path used to calculate the Minkowski sums with the forecasting index [15].

The update function is presented below:

```
INPUTS: agents[0...N], forecastIdx[0...N], agentPositions[0...N], agentSpeeds[0...N],
speedBound,
agentOrientations[0...N], agentDelays[0...N], agentPaths[0...N], boundingVol-
umes[0...N]
WHILE agents NOT EMPTY
a <- agents[0]
removeFromQueue(a)
currentPath <- minkowski(agentPaths[a], boundingVolumes[a], forecastIdx[a])
FOR o IN 0...N
otherPath <- minkowski(predictPath(agentPositions[o], agentSpeeds[o],
agentOrientations[o], forecastIdx[a]), boundingVolumes[o], forecastIdx[a])
IF intersects(currentPath, otherPath) THEN
slowdown <- calculateSlowdown(agentPaths[a], agentSpeeds[a], intersectionPoint,
intersectionTime)
IF slowdown < speedBound THEN
IF isOtherPathAvailable(a) THEN
previousPath <- agentPaths[a]
agentPaths[a] <- calculateNewPath(a)
delay <- calculatePathDelay(agentPaths[a], previousPath)
agentDelays[a] <- agentDelays[a] + delay
addToQueue(a, agentDelays[a])
ELSE
delay <- calculateSpeedDelay(agentSpeeds[a], slowdown)
agentSpeeds[a] <- slowdown
agentDelays[a] <- agentDelays[a] + delay
addToQueue(a, agentDelays[a])
END
ELSE
agentPositions[a] <- updatePosition(agentPaths[a], agentSpeeds[a])
END
END
END
agents <- buildPriorityQueue(agentDelays)
```

4 Concluding Remarks and Future Work

athfinding is a critical element of AI in many modern applications like multiple mobile robots, game industry and flock traffic navigation based on negotiation (FTN).

We develop a new algorithm capable of planning paths for multiple agents on partially known and changing environments inspired by cooperative A* and D*.

From a distributed approach (Decoupled) our collective pathfinding in dynamic environments algorithm decomposes the task of individual plan into weakly-dependent problems for each agent. Each agent can search greedily for a path according to its destination, given the current state of all other agents. Then based on a space-time search space each agent attempt to make a reservation on (x,y,t,δ) where x,y are in the Euclidean space, t is a time measure and δ is a forward planning-vision measure (forecasting index).

Because these kinds of algorithms are problem dependent we developed a modification of our collective pathfinding in dynamic environments algorithm in the FTN context.

Taking care that in FTN the main issue is that it is based on negotiation a conflict-solver module that has the social rules within.

Evidently in FTN we will not in general obtain a globally optimal path from the individual agent perspective but it is a at least a better plan compared with traveling alone (the worst scenario is if an agent can't find Nash or Pareto partners in the whole path so, it becomes a 1-individual flock).

It was shown that the algorithm can, with relatively few modifications, work on continuous domains updated in fixed time-steps, such as those used by most 3D computer games. The shift from using a reservation table to analyzing world geometry allows the work to be cleanly distributed amongst the agents. This creates a clear separation of concerns and the lack of side-effects makes it trivially parallelizable.

We develop a collective pathfinding in dynamic environments in the two representations: A grid structure that wholly describes the traversable game world like the FTN scenario and also into a graph that samples the continuous space over which game agents can move. Future work will be focused on this last issue.

References

1. Astengo-Noguez, C., Sánchez-Ante, G.: Collective Methods on Flock Traffic Navigation Based on Negotiation. In: 6th Mexican International Conference on Artificial Intelligence (MICAI 2007). Springer, Aguascalientes (2007)
2. Astengo-Noguez, C., Brena, R.: Flock Traffic Navigation Based on Negotiation. In: Proceedings of 3rd International Conference on Autonomous Robots and Agents (ICARA 2006), Palmerston North, New Zealand, pp. 381–384 (2006)
3. Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., Thrun, S.: Principles of Robot Motion: Theory, Algorithms, and Implementations. MIT Press, Boston (2005)
4. Dresner, K., Stone, P.: Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism. In: The Third International Joint Conference On Autonomous Agents and Multiagent Systems (AAMAS 2004), New York, July 2004, pp. 530–537 (2004), http://citeseer.ist.psu.edu/630629.html
5. Dresner, K., Stone, P.: Multiagent Traffic Management: An Improved Intersection Control Mechanism. In: The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), Utrecht, The Netherlands, July 2005, pp. 471–477 (2005), http://www.cs.utexas.edu/users/kdresner/pubs/files/2005aamas-dresner.pdf

6. Dresner, K., Stone, P.: Human-Usable and Emergency Vehicle–Aware Control Policies for Autonomous Intersection. In: The Fourth Workshop on Agents in Traffic and Transportation (ATT 2006)., Hakodate, Japan, May 2006, pp. 17–25 (2006), http://www.cs.utexas.edu/users/kdresner/pubs/files/2006att-dresner.pdf
7. Dresner, K., Stone, P.: Sharing the Road: Autonomous Vehicles Meet Human Drivers. Sharing the Road: Autonomous Vehicles Meet Human Drivers. In: The Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India, January 2007, pp. 1263–1268 (2007), http://www.cs.utexas.edu/users/kdresner/pubs/files/2007ijcai-dresner.pdf
8. Latombe, J.C.: Robot Motion Planning, 1st edn., 672 pages. Kluwer Academic Publishers, Springer (1991)
9. Pottinger, D.C.: Coordinated Unit Movement. Game Developer Magazine 3(3) (January 1999), http://www.gamasutra.com/features/game_design/19990122/movement_01.htm
10. Russell, S., Norvig, P.: Artificial Intelligence A Modern Approach, 2nd edn. Pearson Education Inc., New Jersey (2003); Personal Library, ISBN 0-13-790395-2
11. Silver, D.: Cooperative Pathfinding. American Association for Artificial Intelligence (2005)
12. Stentz, A.: Optimal and Efficient Path Planning for Partially-Known Environments, pp. 3310–3315. IEEE, Pittsburgh (1994)
13. Wooldridge, M.: Introduction to Multiagent Systems. John Wiley and Sons, Chichester (2002)
14. Zelinsky, A.A.: Mobile Robot Exploration Algorithm. IEEE Transactions on Robotics and Automation 8 (December 1992)

A Research in Agent Negotiation Forms Dynamic Supply Chain

Jui-Wen Hung[1], Ching-shun Hsieh[2], and Kuo-Yen Lo[3]

[1] Department of Information Management, Ling Tung University
1 Lingtung Rd., Taichung, Taiwan, 408, R.O.C
harng@mail.ltc.edu.tw
[2] Department of Commercial Technology and Management, Ling Tung University
1 Lingtung Rd., Taichung, Taiwan, 408, R.O.C
sjsoon@mail.ltu.edu.tw
[3] Department of Information Management, Ling Tung University
1 Lingtung Rd., Taichung, Taiwan, 408, R.O.C
log@mail.ltu.edu.tw

Abstract. Supply chain management (SCM) aims to efficiently integrate suppliers, manufacturers, warehouses, and retailers, not merely to ensure that merchandise is produced and distributed in the appropriate quantities, to the right locations, and at the right time, but also to minimize system wide costs while satisfying customer requirements. This paper proposes a Dynamic Information Exchange Center (DIEC) for creating a dynamic supply chain network in an Internet environment. Agent technology supports users in negotiating with upstream suppliers or downstream demanders and making decisions regarding partner selection. The framework allows enterprises to find more opportunities to cooperate with other partners. In business, such a framework not only reduces purchase costs but also saves time for enterprises in reaching agreements.

Keywords: SCM, intelligent agent, negotiation, outranking methods.

1 Introduction

The computer and Internet revolution has changed traditional commercial activities, such as shopping, brokerage, negotiating, and retailing [10]. Many enterprises recognize that closed collaboration and rapid information exchange among supply chain members is essential for performing the purchase task in electronic trade environments. Figure 1 illustrates that the traditional environment of supply chain and information transfer is hierarchical. The information flow runs upstream, but the physical flow runs downstream.

Most issues in SCM involve discussing how to reduce inventory and satisfy customer needs. Enterprises also wish to improve their competitiveness by combining new information technology and supply chain management theory. The Internet has become widespread in business-to-business commerce, and enables businesses to

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 869–879, 2008.

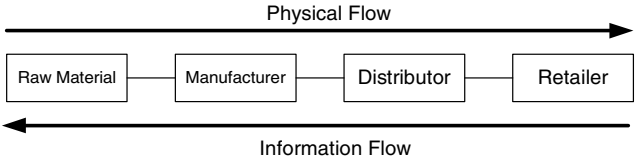


Fig. 1. Traditional supply chain

communicate rapidly and efficiently. However, despite Internet technology being able to assist enterprises in communicating quickly and conveniently, enterprise delegates still need to meet face-to-face to discuss specific business deals. Software agents are personalized, continuously running, and semiautonomous [12][16][13]. Such agents can communicate with other agents through the Internet and perform specific tasks required by the user.

Generally, issues being negotiated are multi-faceted, and for example can include price, quantity, specification, quality, and so on. These attributes must be considered while negotiating process. Most critical situations require experts, known as representatives, to conduct this negotiation. Nevertheless, not all negotiation issues are so complicated that they require negotiator to handle those. Examples of such routine negotiation issues involve negotiations over products like paper, pens, printer ink, and so on.

This paper focuses on purchasing situations that are not directly related to the core business of a firm. An agent is used as a representative to negotiate with other enterprises in accordance with predetermined strategies. Utility function is used to assess opponent proposals and make counter proposals.

2 Literature Review

2.1 Supply Chain Management

Supply chains in modern enterprises comprise a world-wide network of suppliers, factories, warehouses, distribution centers and retailers, through which raw materials are acquired, transformed into products, delivered to customers. To operate efficiently, supply chain functions must work in a tightly coordinated manner [1]. David [6] noted that supply chain integration is difficult for two primary reasons:

- Different supply chain facilities may have different, possibly conflicting, objectives.
- Supply chains are dynamic systems that evolve over time.
- Croom [4] gathered various relevant definitions of supply chain management. Finally, this investigation concludes by discussing some issues related to supply chain management. These issues are:
- Cost reduction: Enterprises typically strive to reduce production costs, including time, stock, human costs, and so on.
- Information sharing: Enterprises must obtain more information, and thus can design appropriate strategies for enterprise benefit.

- Cross-organization integration: Management in a global society no longer involves striving in isolation. Therefore, enterprises must conduct cross organization integration.

2.2 Intelligent Agent

Software agents, which now are very widespread, were first used several years ago to filter information, match individuals with similar interests, and automate repetitive behavior [12]. Agents are computer systems with numerous important capabilities. These capabilities are summarized and described below:

- Autonomy [8]: Users do not need to indicate how to work. Agents can finish task independently.
- Adaptive [1]: Agents can judge environmental change and adjust their task actively. Such agents generally have domain knowledge, and thus can easily identify ideal solutions based on the real conditions.
- Anticipatory [16]: Agents should not only predict user information through task or interaction with environment but also should alert users and react appropriately.
- Cooperation [8]: Agents can communicate with other agents by completing specific tasks together.
- Trustworthy [16]: An agent should satisfy the needs of user by using reliable methods to get user's trust.

Agents can adapt to environmental changes dynamically, and can model variation of management behavior among different managers [17]. Therefore, this work proposes to construct, using agent technology, a supply chain communication and negotiation mechanism for reducing purchase cost.

2.3 Negotiation

Negotiation is a form of decision-making in which two or more parties jointly consider various possibilities with the goal of reaching a consensus [15]. Negotiation is complex because multi-dimension criteria must be considered and action must be prompt.

Rosenschein and Zlotkin [15] proposed a monotonic concession protocol with the following rules:

- Negotiation proceeds in a series of rounds.
- In the first round, both agents simultaneously propose a deal from the negotiation set.
- If one of the agents finds that the deal proposed by the other is as good as or better than its own proposal, it will accept the proposal.
- If no agreement is reached, the negotiation proceeds to another round of simultaneous proposals.
- If neither agent makes a concession in some round, then the negotiation terminates, with the conflict deal.

Faratin [7] designed a negotiation model involving autonomous agents. This multilateral negotiation was based on two parties who exerted a mutual influence on

one another. This paper is based on this concept and considers how to apply this model in supply chain member negotiation and dynamic supply chain formation.

2.4 Outranking Method

Purchasing decisions for supplier selection is strategically important for companies. Borer [2] reviewed existing decision models and concluded that the following properties are worth considering.

- Number of criteria and their nature.
- Interrelatedness of decisions.
- Type of decision rule used.
- Number of decision makers
- Various types of uncertainty.

The outranking method is well suited for making multiple criteria decisions with qualitative as well as quantitative attributes. Numerous outranking methods currently exist, that is AHP, PROMETHE, ELECTRE and so on. No methods are perfect. This paper proposes to use ELECTRE III [3] as a tool for partnership selection. This model is simple and easy to use because users only need to setup some thresholds. This method can prioritize solutions based on user preferences.

3 Dynamic Supply Chain Network

Liautaud and Hammond [11] designed a supply chain extranet for expanding supply chain functionality. This paper proposes a dynamic supply chain model according Liautaud and Hammond's idea, using agent technology to solve the complex integration problem of the dynamic supply chain network. This model should automate and simplify the manufacturing process, and moreover, should allow members at all levels of the supply chain to reach their goals.

3.1 Dynamic Supply Chain Model

In Figure 2, we illustrate a dynamic supply chain model. Retailers, distributors, manufacturers, and raw materials and logistic vendors share information in an Internet environment. The Dynamic Information Exchange Center (DIEC) then gathered the demand information of each member. Suppliers from every level provide their supply conditions and negotiate in the meeting room, while potential buyers evaluate the solutions proposed by opponents and make decisions. Other vendors can gather market movement information and make appropriate production forecasts using the DIEC framework.

3.2 DIEC Framework

The DIEC, depicted in Figure 3, divided into five functions: blackboard, meeting room, database, outranking model, and matchmaker. Any member of DIEC can be a demander. Members can request basic information from other enterprises or post their

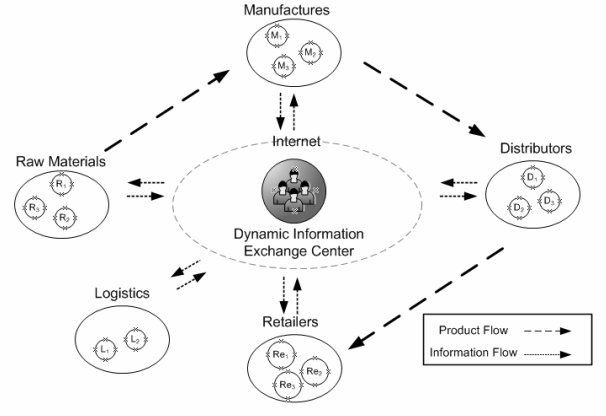


Fig. 2. Dynamic supply chain model

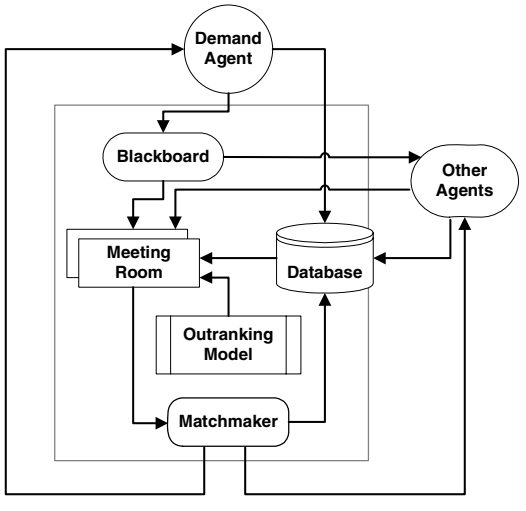


Fig. 3. DIEC Framework

own needs on the blackboard. Other agents then can receive messages from the blackboard. Meeting rooms provide a private environment to support negotiations among agents with matching basic conditions. After potential buyers post their requirements on the blackboard, they negotiate with potential suppliers. During negotiation, the demander agent can use the outranking model to rank solutions and select trading partners. Finally, the decision of the demander will pass to the matchmaker agent who announce the final supply chain network and record it in the database, and notify every member that joins the meeting room.

– Blackboard: Blackboard-based negotiation is derived from the mechanism of negotiation among intelligent agents using a blackboard as a media in which each agent exchanges information [9].

- Meeting room architecture: The meeting room provides a private space for both buyer and seller. The meeting room does not have a physical form. It is a serial of procedures. We will discuss at next session.
- Database: The database records the basic information of each agent, including location, business scope, sale items, and so on. The database provides other members with detailed information on the enterprise they are interested in.
- Outranking model: ELECTRE III uses the pseudo-criterion, with its indifference and preference thresholds, to explicitly make allowances for any imprecision/uncertainty in the data [17].
- Matchmaker: After supply chain formation, the contract is recorded in the database. The matchmaker agent then must confirm the inclusion of each new member in the chain and write into database.

4 Negotiate and Outrank

4.1 Sequence of Demand, Negotiation, and Ranking

Figure 4 (arrow A) shows that demand flows upstream. Assume that demander agent W has potential suppliers M_1 and M_2. Furthermore, assume that M_1 has potential suppliers U_a and U_b, while M_2 has potential suppliers U_c and U_d. When W posts their demand on the blackboard, M_1 and M_2 can decide whether or not they want to trade. Before M_1 proposes its supply solution, it must first check its stock. If M_1 has insufficient stock to supply W, it will then post its own demand on the blackboard.

The negotiation sequence (arrow B) runs downstream. That is, upstream agents first negotiate with midstream agents to achieve agreement at first. Midstream agents then calculate their production costs and draft supply solutions, after which they make proposals to downstream agent. Supposing M_1 has insufficient stock to provide W, and so must request support from upstream enterprises, otherwise, M_1 will handle the demand of W by itself. Each level of negotiations, such as (U_a, U_b, M_1), (U_c, U_d, M_2) and (M_1, M_2, W), uses the same negotiation model to reach agreement. This model is also suitable for use in multi-tier frameworks. If U_a and U_b are interested in the demand of M_1 demand and have sufficient stock to supply M_1, then M_1 will negotiate

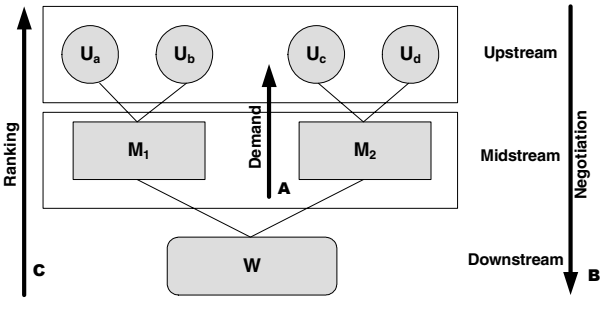


Fig. 4. Proposed sequence of demand, negotiation, and ranking

with U_a and U_b and seek a supply solution for its own supply problem before proposing its supply solution to W. If M_1 has no consensus with U_a and U_b, it will lose the opportunity to negotiate with W. M_2's activity is the same as M_1.

Partner selection (arrow C) differs from the direction of negotiation. When M_1 and M_2 reach agreement with W, W has right to choose which one of these two suppliers will be its partner. M_1 and M_2 then use the same method to select their own upstream vendors.

4.2 Agent Interaction Diagram

Figure 5 describes the situation of agent interaction. Step 1 involves identifying possible supply chain choices. If no supply agent enters the room before the meeting room deadline, then the meeting room manager agent will cancel the meeting. The process proceeds directly to step 8.

Step 2 shows that some supplier agents are in the meeting room. Supply agents present their proposals to demand agents, and demand agents then assess those proposals. If the proposal of supply agent satisfies the needs of the demand agent, agreement will be reached. On the other hand, if the proposal does not satisfy the needs of the demand agent, the demand agent then presents a counter-proposal to the supply agent. These processes continue until agreement is reached, the counter-proposals of one or both parties are exhausted, or the allotted negotiation time is exhausted. Negotiation issues include quantity, price, due date, and so on. If no proposals can satisfy the requirements of the demand agent, the system skips to step 8. In this step, all supply agents negotiate with demand agent individually.

In step 3, the system gathers all of the proposals such that agreement is reached. Those supply agents will be candidates to form supply chain partner. The system can cancel any proposals that agents retract (step 5).

In step 4, the demand agent uses outranking method to select the cooperation partner. The outranking issues include supply agent proposal cost, credit condition, financial state, and so on. After the demand agent has made a decision, the system notifies the relevant supplier and other candidate suppliers. Once the relevant supplier has agreed to deal, the agreement becomes a contract and the meeting ends (step 7), else the deal will be cancelled (step 6).

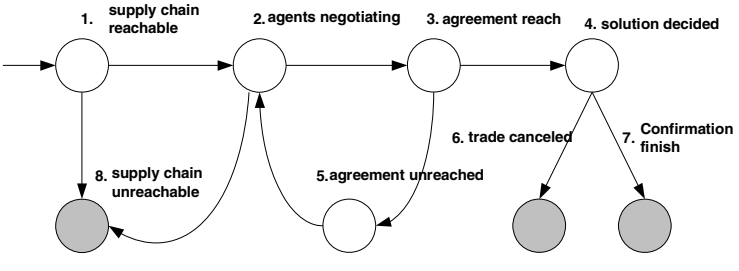


Fig. 5. Agent interaction diagram

5 Outrank Example

As for negotiation issues, we give two assume parameters: price and purchase quantity. Each agent has its own acceptance range for the alternative. For example, Table 1 is the price and quantity acceptance range of M_1 and W.

Table 1. Acceptance range

	Price	Quantity
W	[15, 18]	[800, 1200]
M1	[16, 20]	[900, 1400]

The value function is used to calculate those acceptance ranges. As depicted in Figure 6, each agent calculates value by its utility function.

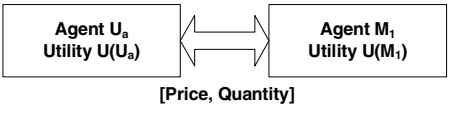


Fig. 6. Each agent calculates value by its utility function

M_1 wishes to negotiate with W. The acceptance range of W regarding price is from \$15 to \$18, denote P[15, 18]; while the acceptance range for quantity is from 800 to 1200, denote Q[800, 1200]. Assume the best benefit alternative for W is [15, 800]. M_1's acceptance range for price and quantity is [16, 20] and [900, 1400] respectively. Assume the best benefit alternative for M_1 is [20, 1400]. Furthermore, the preference weight of W for price and quantity, respectively, are 0.6 and 0.4; and the preference weight of M_1 for price and quantity, respectively, are 0.4 and 0.6. According to the above base data, a value can be calculated for each round [price, quantity] alternative using the value function. The calculated result is called the utility value. For example, assume one of the alternatives that M_1 propose is [17, 900]. This alternative can be calculated as a value for W, as follows:

$$\frac{(17-15)}{(18-15)}\times 0.6+\frac{(900-800)}{(1200-800)}\times 0.4=0.5$$

Therefore, after calculating all possible alternatives, M_1 can rank these alternatives from high to low. W also can rank its own alternatives. Both sides propose alternatives based on their own preference sequences. When M_1 proposes an alternative to W, this alternative can be calculated as a utility value by W. If this utility value is less than the alternative W is planning to propose in the next round, W will not accept this alternative, and instead propose its own preference alternative to M_1. M_1 then calculates utility value of the alternative of W, and makes a decision. If the alternative utility value of W exceeds the utility value of that which M_1 is planning

Table 2. Performance Table

	Cost	Credit (1~100)	Quality (1~100)
M_1	160,000	82	90
M_2	165,000	90	85

Table 3. Three kinds of thresholds and criteria weights

	Cost	Credit	Quality
Indifferent Threshold	2,000	2	3
Preference Threshold	5,000	5	5
Veto Threshold	10,000	10	10
Weight	0.5	0.3	0.2

to propose in the next round, then M_1 will accept the proposal of W. In this case agreement is reached.

While M_1 and M_2 both reach agreement with W, W calculates the purchase costs of each supplier and establishes a performance table as Table 2. The credit means the degree that a vendor be trusted by other supply chain partner. Moreover, quality represents whether the state of the product is good or bad. Credit and quality are all established based on the recognition of W. If W prefers a particular enterprise, it will assign that enterprise a high degree of trust. These evaluations can be established prior to negotiation.

Table 2 illustrates that W finished negotiating with M_1 and M_2 and make a performance table. W provides three thresholds, namely an indifferent threshold, a preference threshold, and veto threshold, as well as the criteria weight. The three thresholds illustrate user's feelings regarding each item listed in the performance table.

According to the above data, W ranks those two suppliers by using the ELECTRE III outranking method. The result is that M_2 becomes more suitable for W than M_1. This information is received by the matchmaker, and the matchmaker then notifies each of the members involved in the negotiations and confirms the agreement.

6 Conclusion

This paper focuses on how to build a dynamic supply chain through agent negotiation. The negotiation mechanism and outranking methods assist agents in gathering quantification and non-quantification conditions for selecting collaboration partners. During the negotiation process, the agent serves as a tool that estimates opponent supply conditions based on user presetting of their value function. Meanwhile, value function is a mathematical formula that represents the preferences of each participant in different negotiation conditions. ELECTRE III can sort candidates according to demand preferences. The demander need only set their preference, indifference, and veto thresholds, and can use these parameters to sort potential suppliers. This

automatic mechanism not only reduces purchase costs but also saves time for enterprises in reaching agreements.

We think this paper makes three key contributions, as follows:

- Increase information transparency:
- Uniform negotiation model
- Consider quantification and non-quantification criteria

In practice, this model is suitable for automating the task of work-related material purchasing. Enterprise negotiators only need to consider primary raw materials purchasing, and can let agents handle the purchase of other work-related materials. This approach can save purchase time and costs. Additionally, a dynamic supply chain is formed to satisfy enterprise demand, to prevent it being frustrated by a single supplier running out of stock.

References

1. Barbuceaunu, M., Teigen, R., Fox, M.S.: Agent Based Design and Simulation of Supply Chain Systems. In: WET-ICE 1997, pp. 36–41. IEEE Computer Society Press, Los Alamitos (1997)
2. Borer, L.D., Wegen, L.V.D., Telgen, J.: Outranking Methods in Support of Supplier Selection. European Journal of Purchasing and Supply Management 4, 109–118 (1998)
3. Buchanan, J., Phillip, S., Daniel, V.: Project Ranking Using ELECTRE III, Department of Management Systems University of Waikato, Hamilton, New Zealand, Research Report Series (1999)
4. Croom, S., Romano, P., Giannakis, M.: Supply Chain Management: An Analytical Framework for Critical Literature Review. European Journal of Purchasing and Supply Management 6, 67–83 (2000)
5. David, P., Peter, S.: Predictive Planning for Supply Chain Management. In: Proceedings of the International Conference on Automated Planning and Scheduling, Cumbria, UK (2006)
6. David, S.L., Philip, K., Edith, S.L.: Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies. McGraw-Hill, New York (2000)
7. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation Decision Functions for Autonomous Agents. International Journal of Robotics and Autonomous Systems 24, 159–182 (1998)
8. Genesereth, M.R., Ketchpel, S.P.: Software Agents. Communication of the ACM 37, 48–53 (1994)
9. Ito, T., Salleh, M.R.: A Blackboard-Based Negotiation for Collaborative Supply Chain System. Journal of Materials Processing Technology 107, 398–403 (2000)
10. Kraus, S.: Strategic Negotiation in Multiagent Environments. MIT Press, Cambridge (2001)
11. Liautaud, B., Hammond, M.: e-Business Intelligence: Turing Information into Knowledge into Profit, pp. 244–245. McGraw-Hill, New York (2000)
12. Maes, P.: Agents that Reduce Work and Information Overload. Communication of the ACM 37, 31–40 (1994)
13. Minghua, H., Rogers, A., Xudong, L., Jennings, N.R.: Designing a Successful Trading Agent for Supply Chain Management. In: Proc. 5th Int. Conf. on Autonomous Agents and Multi-Agent Systems (2006)

14. Rogers, M., Bruen, M., Maystre, L.Y.: ELECTRE and Decision Support: Methods and Applications in Engineering and Infrastructure Investment. Kluwer Academic, Dordrecht (2000)
15. Rosenschein, S.J., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation among Computers. MIT Press, Cambridge (1994)
16. Sycara, K., Decker, K., Pannu, A., Willliamson, M., Zeng, D.: Distributed Intelligent Agents. IEEE Expert (1996)
17. Yung, S.K., Yang, C.C.: Intelligent Multi-Agents for Supply Chain Management. In: IEEE SMC 1999 Conference Proceedings, pp. 528–533 (1999)

BROA: A Bayesian Robotic Agents Architecture

Fidel Aznar, Mar Pujol, and Ramón Rizo

Departamento Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Campus de Sant Vicent s/n, 03080, Alicante, Spain
fidel@dccia.ua.es, mar@dccia.ua.es, rizo@dccia.ua.es

Abstract. Autonomous intelligent agents paradigm has encouraged robotic researches to take another step forward in the design of control architectures replacing modules with agents. This paper presents a logical fusion between Bayesian theory and artificial intelligent agents, showing a new intelligent Bayesian agent architecture oriented towards Bayesian robotics. To define this architecture we will provide a common framework for developing intelligent agent applications using Bayesian theory. We will also review some of the most important Bayesian agent applications and we will compare them with our model. Finally, a simple robotic application will be provided as a proof of concept of the presented architecture.

Keywords: Bayesian Agents, Autonomous Intelligent Agents, Bayesian Programming.

1 Introduction

In recent years, probabilistic algorithms have become very popular in mobile robotics because they explicitly represent robot's uncertainty, using a robust mathematical background. A robotic agent could use a model of the environment representing the real universe where it will interact. Nevertheless, any model of a real phenomenon will always be incomplete due to the existence of unknown, hidden variables that will influence the phenomenon, causing the model and the phenomenon to have different behavioral patterns. Probabilistic inference helps to solve this problem taking into account incomplete and uncertain information. In addition, probabilistic approaches are usually more robust than their traditional counterparts [1].

Different efforts [1][2][3] have been made to define programming frameworks for Bayesian programming, motivated by successful probabilistic methods for mobile control with the main goal of facilitate the development of such probabilistic software. However, most probabilistic frameworks focus only on structural or object programming.

Some recent work with agents and multi-agent systems has encouraged robotic researches to take another step forward in the design of control architectures replacing modules with agents. Although some agent and multi-agent applications that use Bayesian inference [4][5][6][7][8] exist, we have not found any architecture or

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 880–888, 2008.

framework for agent design, centered on Bayesian programming, that fully exploits its benefits. Bayesian programming agents will focus on discrete or continuous joint distributions as a basic information chunk for developing all the principles that an autonomous intelligent agent must follow.

In this paper a probabilistic agent framework, oriented to robot programming, will be provided. Firstly, we will review the most important design issues for developing good autonomous intelligent agents. Secondly, a definition and a specification for Bayesian robotic agents will be provided and also discussed with the previously commented design issues. Next, our framework will be briefly compared with recent agent architectures that use Bayesian inference. Moreover, a conceptual robotic example will be formalized using the proposed framework. Finally, conclusions and future lines will be commented.

2 Agent Design Issues

Although an established theory regarding autonomous intelligent agents does not exist, and designing autonomous agents is an active research area, there are several principles that could be used to guide us for the autonomous agent design [10]. These principles may help researchers to obtain better artificial intelligence systems. We will review some of the most important principles and we will use it in the next section to propose our framework.

Autonomy. A perfect agent should work with no human supervision o instruction, being self-sufficient.

Emergence. When programming an agent, the programmer has a specific idea about the task to be solved by this agent. However, to design an agent according to his view of the task may not always be the best solution because his view of the task depends on human perspective. The agent may have a different embodiment and then a different perspective of the task. For example, one solution to these problems arises when the programmer designs the system for emergence (when behaviours that are not programmed results from agent-environment interaction and from self-organization of the agent's control system).

Epigenesis. This term defines a process through which increasingly more complex cognitive structures emerge in a system as a result of interactions with the physical and social environment. The final behaviors are consequence of environment information and existing information. In this way the role of the environment is constructive rather than selective. Moreover, new tasks should be learned without requiring a redesign of the control system, where human teachers may affect this learning process only as a part of the environment without interfering with its internal representation.

Parallel, loosely coupled processes. Loosely coupled processes and little or not centralized resources are needed for supporting emergence. When the control is decentralized and distributed, intelligent behaviors may emerge from the linked dynamics of the number of processes, each of which contributes to the overall function, as the agent interacts with the environment.

Sensory-motor coordination. In classical AI the cycle perception-think-action explicitly separates time perception from action, preventing the potential emergence of adaptive behaviors from their coordination. A continuous interaction between perception and action should exist, where perception guides action in interaction with internal state and action changes internal state and the perspective of the environment as perceived by the agent. Moreover, as stated in [9], cognition and learning need perceptual and action capabilities to be developed.

Goals. An agent must be goal directed and must have a value system that would guide its behaviors. It must have predefined drivers that would induce the exploration of the environment. This exploration may lead to non-trivial sensory motor patterns, as a consequence of self-organization.

Internal representation. The action an agent performs should not be only determined externally, based on inputs, as in purely reactive systems, but neither be planned ahead, as in some classical AI systems. This representation should not be imposed by the user or the designer of the system, but should grow in the sensory motor interaction of the agent with the environment.

Symbolic Communication. Communication between agents requires interchanging symbols that must be understood by all the agents in communication. The meaning of these symbols should not be given by agent programmer but as in previous principle should grow in the sensory motor interaction with the environment. Otherwise, the meaning cannot be accessible to the agent itself.

3 Bayesian Robotic Agents

In this section we will define an internal schema that provides support for Bayesian agents oriented towards robotic programming, briefly comparing our approximation with those proposed in [4][5][6][7][8].

Intelligent Bayesian Agents are a group of agents, designed for the formalization of complex robotic systems, which are based on probabilistic theory. A pure Bayesian agent is an agent that only uses Bayesian Inference for cognition and learning and its internal information is composed exclusively by joint probabilistic distributions. Without lost of generality we will center this section with pure Bayesian agents, as it is easy to extend a pure Bayesian agent with another kind of internal information or learning mechanism. However, we have to underline that a wide range of robotic applications could be solved using only Bayesian tools [2][3].

In figure 1, the internal schema of a Bayesian agent is presented. This kind of agents uses random variables and joint probabilistic distributions to model every step of perception, cognition and action.

Firstly, all variables must be specified and classified in one of the following groups: goals, behaviors, internal states, communications, perceptions and actions. Obviously more than one variable can be specified inside a group and one group could be empty. This classification shows the most important design issues presented in the previous sections. For example, communication variables specify the information symbols (Bayesian variables or distributions) to be emitted or received; state variables organize the internal representation of the agent.

$$
\begin{cases}
\textbf{BAYESIAN AGENT} \\
\text{Identify agent information} \\
AgentInformation = \begin{cases} Goals, Behaviors, States, Communication \\ pErception, Action \end{cases} \\
\text{Propose joint distribution decomposition of } AgentInformation \\
\begin{cases} P(G,B,S,C,E,A) = \begin{cases} 1)\ P(G \mid S) \times P(B \mid G,S) \times P(S \mid B,C,E,A) \times P(C \mid S) \times \\ \qquad \times P(A \mid E,B) \times P(E) \\ 2)\ P(G) \times P(B \mid G) \times P(S \mid B,E,A,C) \times P(A \mid B,E) \times \\ \qquad \times P(C \mid S) \times P(E) \\ \text{Previously based} \end{cases} \end{cases} \\
\forall i, i \in AgentInformation, \text{ Define the 'use' of each variable} \\
\begin{cases} \forall \text{question, question} \in \text{i} \\ \text{Solve the question using: } \underbrace{P(U \mid K) = \alpha \sum_{H} P(H,U,K)}_{\substack{\text{where } H \text{ are hidden } U \text{ unknow variables} \\ \text{and } K \text{ know variables. } \alpha \text{ is a normalization term}}} \end{cases} \\
\text{Use the distribution} \\
\text{Specify Bayesian inference method to be used}
\end{cases}
$$

Fig. 1. Internal schema that defines a Bayesian Agent

Secondly, the joint probabilistic distribution of all variables should be specified. This helps to calculate the 'use' of each variable. If the distribution is fully identified any question related to it could be asked. These variables decomposition shows the programmer knowledge about the problem but do not fix the behavior of the agent because usually some terms of the distributions must be learned. Bayesian learning allows the agent to increase more complex cognitive structures, enabling emergent behaviors and epigenesis. Two decompositions are proposed as a guide:

- The first decomposition reflects the interactions required for the basic design principles commented previously: in order to allow sensory-motor coordination the perception of the agent influences their actions in a reactive way. But actions are not only influenced by perceptions but also by behaviors, which raise the system from reactive level to hybrid level. Communications between agents require interchanging symbols that should be obtained from sensory-motor interaction, therefore communication variables influence the internal state of the robot. This state is obtained from the interaction of perception and actions, the agent communication and the behavior variables. In the same way, goals are driven by states, which allow changing dynamically its goals.
- The second one is a simplified version but easier to use for Bayesian Inference (because this decomposition could be modeled and solved using Bayesian networks tools). In this case the agent cannot change its initial goals and actions are also not affected directly by a set of behaviors, only indirectly by agent states, which lead to a more simplified system that reduces the possibilities of emergence, intelligence and epigenesist, but still allows them.

- Next, for each variable defined in the system, its 'use' must be specified. We define the 'use' as a set of questions that the agent must answer. These questions could be always specified using both, the full joint probabilistic distribution and the general inference: $P(U \mid K) = \alpha \sum_{H} P(H,U,K)$. Once the question is specified all the terms must be defined, or approximated using discrete or continuous distributions, or even learned.
- Finally, the programmer must specify the Bayesian inference method to be used in the agent and the way the information obtained from the distribution is used.

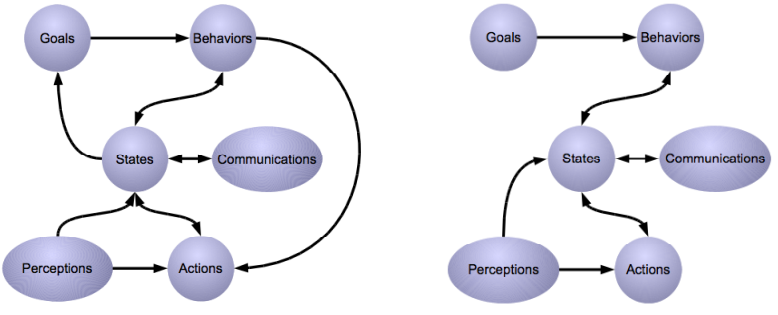


Fig. 2. Bayesian dependences between agent variables. Left figure represents these dependences taking into account the design issues previously commented. Right figure relaxes such assumptions for simplifying Bayesian inference providing a Bayesian Network as decomposition.

3.1 Comparing with Other Architectures

Now, we will present a brief review of the state of the art of Bayesian agents in scientific literature, comparing it with our proposed architecture. Three groups of applications will be presented as representative parts of this field.

The first group is related to other Bayesian agents frameworks. Santos, Fagundes and Vicari present in [8] an ontology for Bayesian agents interoperability. Moreover, a Bayesian agent architecture is presented. This paper shows a general ontology for working with Bayesian Networks. Compared with our proposed model, this architecture does not provide any information about how defining Bayesian agents and what principles must be used for their design, it is only focused on how Bayesian agent will interact with its Bayesian Network ontology. This Bayesian architecture can be used with our model in case of the result of the variables decomposition will be a Bayesian network, providing a tool for agent interoperability. We have not found any general agent architecture that uses Bayesian Theory guiding the user for agent design, problem solving and robotic programming.

The second one is related to purely Bayesian agents that are defined adhoc for a specific problem, solving it without using an agent architecture. One of these applications is related to coordination and coalition formation in multiagent systems [4][5][6]. In these papers the Bayesian tool used for information decomposition and learning are based on MDP (Markov decision process). A good example of Bayesian

Inference is presented but no general framework for agent design is provided. However, these applications are pure Bayesian and thus could be modeled using our proposed architecture.

The third group is related to Bayesian agent applications, such as [7], where Bayesian tools are used for learning only as a part of the agent. The advantages of Bayesian Probability are only obtained in a limited way.

We want to emphasize the most important advantages of our proposed architecture, most of them derived from both, artificial agents and Bayesian theory:

- This architecture provides a common framework for comparing different Bayesian agent
- It supports the most important design principles in intelligent agent systems
- Agents can use incomplete and uncertain information in a direct way when using the proposed model
- A wide range of Bayesian inference tools could be used for variable learning or marginalization
- Finally, despite all these advantages, not all problems are suitable for being modeled using this framework, because Bayesian inference is, in general, a NP-hard problem. For this reason programmers must choose a distribution decomposition taking into account both, agent design and computational complexity.

4 A Proof of Concept Example

Imagine a robotic agent that must perform the following actions: when the agent detects a source of light it must move to it. Moreover, it must avoid obstacles using its sonar sensors. In this section we will review a Bayesian agent, presented in figure 3, that performs these actions.

Once we identify agent variables and proposed join probabilistic distribution decomposition for the problem, we need to specify the use of each variable. Related to goals, communications and perceptions there is no question that the agent needs to know (for example, in this case the agent must use a fixed set of parallel goals, so we do not need to ask which goal will be used next).

The utility of behavior variable B is defined with the question $P(B \mid G,S)$. As the behaviors defined in this distribution are specified directly based on the agent goals modifying the distribution in a uniform way (we do not prefer any of them more than the other), we could omit these goals and only consider the internal state of the agent for obtaining the distribution of the next behavior. For example this information could be useful in the case of communicating to another agent the influence of phototaxy or obstacle avoidance in a specific place of the environment.

For obtaining which action must be sent to the robot we could think to ask the question $P(A \mid E)$, which leads to a pure reactive system. We need to take into account the two behaviors proposed, so we include the behavior variable B using the marginalization rule. Finally, we decompose $P(A,B \mid E)$, establishing that $P(A \mid E) = \sum_{B} \left(P(B \mid E) \times P(A \mid E,B) \right)$. The first term of the distribution will obtain the probability of activating phototaxy or obstacle avoidance depending on the actual

$$
\begin{cases}
\textbf{AGENT Phototaxy and AvoidObstacles} \\
\text{Identify agent information} \\
AgentInformation = \begin{cases}
Goals: G = \{showPhototaxy, showColisionAvoidance\}; \\
Behaviors: B = \{phototaxy, colisionAvoidance\} \\
States: S = \{B_{t-1}\}; \\
Communication: \varnothing; \\
Perception: E = \{s_1, s_2, ..., s_7, photo_1, photo_2\}; \\
Action: A = \{vrot, vtras\}
\end{cases} \\
\text{Propose joint distribution decomposition of } AgentInformation \\
\{P(G,B,S,C,E,A) = P(G) \times P(B \mid G,S) \times P(S \mid B,C,A,E) \times P(A \mid S,E) \times P(C \mid S) \times P(E) \\
\forall i, i \in AgentInformation, \text{ Define the 'use' of each variable} \\
\begin{cases}
G, C, E = \{\varnothing\} \\
B = \{P(B \mid G,S) = P(B \mid B_{t-1})\}; \\
S = \{P(S \mid B,C,A,E) = P(S \mid B,E)\} \\
A = \{P(A \mid E) = \sum_B P(A,B \mid E) = \sum_B \left(P(B \mid E) \times P(A \mid E,B)\right)
\end{cases} \\
\text{Use} \\
\begin{cases}
1) \text{ Ask actual behavior with: } P(B \mid B_{t-1}) \\
2) \text{ Determine the action to develop: } \sum_B \left(P(B \mid E) \times P(A \mid E,B)\right) \\
3) \text{ Redefine previous state with: } P(B_{t-1} \mid B,E)
\end{cases} \\
\text{Statistic learn of } P(A \mid E,B)
\end{cases}
$$

Fig. 3. Simple Bayesian Agent for obstacle avoidance and phototaxy behavior

perception. Once the probability of each behavior is defined, we use the second term with the actual perception to determine which action must be sent to actuators.

Next, the use of each question must be defined. We first obtain the distribution of the actual behavior using the previous one. This information is not used in this simple example but, as commented before, could be used to communicate other agents the influence of both behaviors in a specific place. Then, we obtain the action to be sent to the robot using the influence of each behavior (determined by perception variable). Finally we review the previous state regards the actual state and perception. This is a smooth Bayesian filter that uses the perception and the actual state to recalculate the probability of being in the previous state (it provides a better estimation of the previous state, which could be used to determine the current state, because we have more data available).

All these distributions and decompositions are defined by the programmer (with their knowledge about the problem). Therefore, until now, we provide a fixed design that not accomplishes epigenesis design principle. But even in this simple example we will see how we can achieve different behaviors using learning with the same agent structure.

We state previously that we need to solve $\sum_B \left(P(B \mid E) \times P(A \mid E,B)\right)$. The first term is easy defined using a table where each perception defines the intensity of each behavior (if we are close to an obstacle *AvoidObstacles* will have much more probability than in empty space; in the same way if we see a brighter light we will go towards it with more intensity than if the light is smooth). But, how can we obtain which action

to perform given the actual perception and behavior? How can we mix both behaviors? There is an easy solution to solve both problems: learning.

One way for learning this distribution is to control the robot using a joystick. When we control the robot performing phototaxy and collision avoidance behaviors, we can collect for each perception which action is preferred. In this probabilistic framework we can learn a Gaussian distribution for each pair perception/action. As they are conditional independent, we can learn in separate ways the phototaxy behavior and the obstacle avoidance behavior to specify the distribution $P(A \mid E,B)$. Even in this simple example, this model provides some advantages if we compare it with other Bayesian Agents systems: the programmer specifies the way the information is used but do not fix the system for doing a specific task. Although we learn phototaxy behavior, we could easy train the system to perform a different action such as to avoid light without changing a line of code (we only must train the robot, moving it with the joystick and performing a light avoid behavior). Using this model emergence and epigenesis is supported by the system in a natural way. Obviously, a more complex multiagent system will provide a way to evaluate how using the proposed agents will support emergence, parallel processes and communication, but this is out of the scope of this paper.

5 Conclusions and Future Lines

In this paper an agent architecture that joins intelligent agents and Bayesian programming is presented. This architecture provides a common framework to design intelligent agents that use Bayesian inference. Agents defined with this model could work with uncertain and incomplete information in a well-established mathematic framework. In addition, this architecture also supports the most important design issues for intelligent autonomous agents. Moreover, we compare our agent architecture with other Bayesian agents and find no other model that provides a general framework for specifying Bayesian agents, not only for robotic programming but also for general cognitive agents. Furthermore, in this paper a robotic application has been provided and modeled as a proof of concept of our agent architecture. Future lines will be related to analyze specific agent decompositions that take into account Bayesian inference method for reducing computational requirements. We are also modeling a robotic multiagent system using Bayesian agents where coordination and emergence are our investigation priorities.

Acknowledgements

This work has been financed by the Generalitat Valenciana project ARVIV/2007/071.

References

1. Thrun, S.: Towards Programming Tools for Robots that Integrate Probabilistic Computation and Learning ICRA, pp. 306–312 (2000)
2. Park, S., Pfenning, F., Thrun, S.: A probabilistic language based upon sampling functions SIGPLAN, vol. 40, pp. 171–182. ACM, New York (2005)

3. Lebeltel, O., Bessière, P., Diard, J., Mazer, E.: Bayesian Robot Programming Auton. Robots, vol. 16, pp. 49–79. Kluwer Academic Publishers, Dordrecht (2004)
4. Chalkiadakis, G., Boutilier, C.: Coordination in multiagent reinforcement learning: a Bayesian approach. In: AAMAS 2003: Proceedings of the second international joint conference on Autonomous agents and multiagent systems, pp. 709–716. ACM, New York (2003)
5. Chalkiadakis, G., Boutilier, C.: Bayesian Reinforcement Learning for Coalition Formation under Uncertainty. In: AAMAS 2004: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1090–1097. IEEE Computer Society, Los Alamitos (2004)
6. Chalkiadakis, G., Markakis, E., Boutilier, C.: Coalition formation under uncertainty: bargaining equilibria and the Bayesian core stability concept. In: AAMAS 2007: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, pp. 1–8. ACM, New York (2007)
7. Ueno, M., Okamoto, T.: Bayesian Agent in e-Learning ICALT, pp. 282–284 (2007)
8. Santos, E.R., Fagundes, M.S., Vicari, R.M.: An ontology-based approach to interoperability for Bayesian agents. In: AAMAS 2007: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, pp. 1–3. ACM, New York (2007)
9. Steels, L., Kaplan, F., McIntyre, A., Looveren, J.V.: Crucial factors in the origins of word-meaning. In: Wray, A. (ed.) The Transition to Language. Oxford University Press, Oxford (2002)
10. Florian, R.V.: Autonomous artificial intelligent agents Center of Cognitive and Neural Studies (CONEURAL) (2003), http://www.coneural.org/reports/Coneural-03-01.pdf

Chaos and Budworm Dynamics of Agent Interactions: A Biologically-Inspired Approach to Digital Ecosystems

Gabriel Alejandro Lopardo[1] and Fátima N. Rateb[2]

[1] ARLab Agents Research Lab, IIiA Institute of Informatics and Applications, EPS-P4 Campus Montilivi, University of Girona, E-17071 Girona, Spain
glopardo@eia.udg.edu
[2] CASA Centre for Advanced Spatial Analysis, University College London, 1 - 19 Torrington Place, London WC1E 7HB, UK
f.rateb@ucl.ac.uk

Abstract. A primary motivation for research in digital ecosystems is the desire to exploit the self-organized properties of natural ecosystems. Ecosystems are thought to be robust, scalable architectures that can automatically solve complex, dynamic problems. However, the biological processes that contribute to these properties have not been made explicit in digital ecosystem research. Here, we discuss how biological properties contribute to the self-organized features of natural ecosystems. These properties include populations of evolving agents, a complex dynamic environment, and spatial distributions which generate local interactions. The potential for exploiting these properties is then considered. Theoretical and conceptually inspired from nature, the digital ecosystem could behave at large complex cycles of stability through alternate stable states. These approaches suggest that incorporating ideas from theoretical ecology can contribute to useful self-organized properties in digital ecosystems.

Keywords: digital ecosystems, budworm dynamics, chaos, complexity, bio-inspired computation.

1 Introduction

The term Digital Ecosystem (DE) has been used to describe a variety of concepts. Some of these refer to the existing networking infrastructure of the internet [1], [2]. Several companies offer a Digital Ecosystem service, which involves enabling customers to use existing e-business solutions [3], [4], [5]. However, perhaps the most frequent references to Digital Ecosystems arise in Artificial Life research. These DEs are created primarily to investigate aspects of biological and other complex systems [6], [7], [8], [9], [10], rather than to provide a service for human users. A digital ecosystem is a loosely coupled, domain clustered, demand driven collaborative system with an environment where each digital (agent) species is proactive and responsive for its own benefit or profit. It is a self-organized infrastructure with emergent properties aimed at creating an open environment for networked organisations or agents.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 889–899, 2008.

The open environment supports cooperation, knowledge sharing and development of open and adaptive technologies and evolutionary domain knowledge rich environments through dynamic interactions within and between different domain clusters and their environment and collaborative effort to remain balanced in the ecosystem. In a broader sense, it is also a social network model in the digital economy.

Digital Ecosystems with its systemic approach, which empowers small and medium-sized enterprises (SME's) and allows them to participate actively in the global economy, exploiting all dimensions of dynamic collaboration and networking: networking organizations, networking services, networking knowledge and capacities. Digital ecosystems transcend the traditional, rigorously defined, collaborative environments from centralised, distributed or hybrid models into an open, flexible, domain cluster, demand-driven, interactive environment. A digital ecosystem is a newly networked architecture and collaborative environment that addresses the weakness of client-server, peer-to-peer, grid, and web services. The advantages of digital ecosystems are founded on their analogy to ecological systems, nature-inspired architecture.

The focus of our approach is on autonomous systems, namely intelligent agents, who inherit all the problems mentioned above. Their interactions will be grounded in an ecosystem of agents, where agents will be self-organised. One important question to analyse these Digital Ecosystems is *the inclusion of interactions of the agents*. The interactions always exist as for example, the agents that trade in electronic commerce have to manage finally the negotiation for product delivery to the person or institution that these agents represent.

A known Digital Ecosystem, introduced recently, is the Digital Business Ecosystem (DBE) project [11]. The DBE represents a business interaction concept supported by a software platform that is intended to have the desirable properties of natural ecosystems. In the framework of Digital Business Ecosystems, (DBE), an EU IST FP 6 integrated project investigating the design of a digital business ecosystem, which is the enabling technology for the Business Ecosystem; the Open Negotiation Environment (ONE) project allows organizations to create contract agreements in order to supply complex, integrated services as a virtual organization (or coalition). The ONE project is a STREP project funded by the European Commission under the DG-INFSO that is geared towards Small and Medium Enterprises (SME's) in order to provide them with a trusted, secure and free of charge technological environment in which they are able to create tactical and strategic alliances; for the ultimate goal of pursuing business opportunities and growth.

An Open Negotiation Environment is defined as evolutionary self-organising system aimed at creating a digital software environment for small (virtual) organisations that support the regional and local development by empowering open, distributed and adaptive technologies and evolutionary business models for small organisations growth. An open-source distributed environment will support the spontaneous evolution, adaptation and composition of agent communities.

This involves taking ideas, concepts, and designs from biology and applying them to computing. In engineering terms, this would be described as discovering biological design modules [12] to be used in computer systems. In computing, the best known example would be Evolutionary Computing, where the concepts of evolutionary theory are used to solve optimisation problems, including: Agent-based modelling of complex systems; novel forms of distributed evolutionary computing for solving

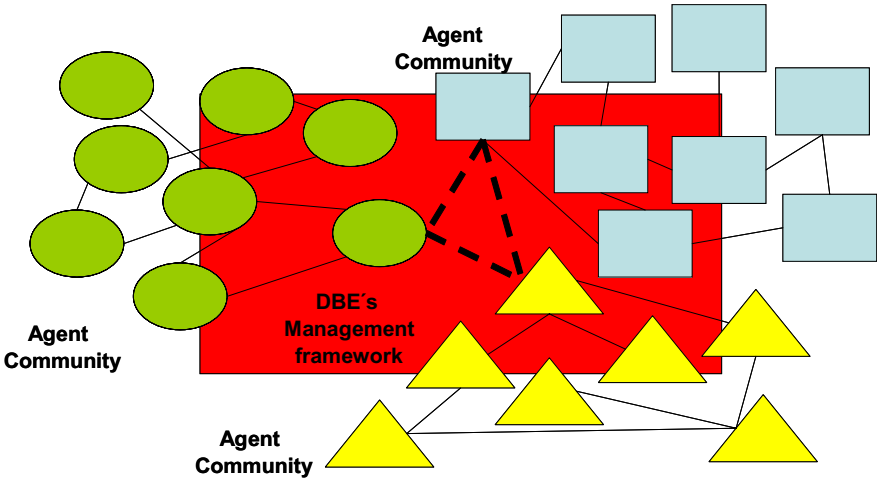


Fig. 1. Groups of intelligent, autonomous and heterogeneous agents, which have different characteristics, and which interact with each other under a set of diverse behaviours ("Styles"), with the purpose of satisfying individual goals and/or common goals

optimisation problems; and neural networks, which can be used to provide a learning-based intelligence which is suitable for a wide variety of applications. The next step would be to develop Biological Design Modules (i.e. agent communities) from ecological and biological systems, and adapt their application in software engineering (Figure 1). The ultimate goal would be to create a reference library of Biological Design Modules which could be used in computing. So, the first step will be to identify the characteristics of biological modularity that are applicable to computing, specifically to Digital Business Ecosystems.

The current scene in computer science is characterized by a growing interest in the ecological-embodied-enactive approach [13], [14]. According to this view agent cognition is best characterized as belonging to embodied, situated agents. In the digital ecosystem would be possible implications of this complex approach for future digital systems. In particular, the enactive approach in perception would lead to agents whose ability to perceive not only depends on, but is constituted by, their possession of certain sensor skills [15], [16]. Noë argues that perception and perceptual consciousness depend on capacities for action and thought, that perception is a kind of thoughtful activity. In the future will be necessary to consider the question to what extent an autonomous (embodied and situated, also) agent could develop its own subjective point of view; and how the users will cope with that trait.

Ecologists increasingly recognize that in nature the indirect coupling of different ecosystem types (e.g. ocean and coast communities, ponds and terrestrial communities, benthic and pelagic communities) through fluxes of organisms across ecosystems boundaries can have strong impacts on the dynamics and structure of communities [17], [18], [19]. Werner [19] estimated that more than 80% of all animals have complex life cycles, in which there is a major transition in the ecology between two ontogenetic life stages, i.e. an ontogenetic niche shift [20]. As a

consequence, different stages within a species often occupy different ecosystems or environments and consume different resources [21], [22]. Although the change in ecosystem or environment use is obvious, it is also very common. Recent studies indicate that organisms with complex life cycles can promote strong trophic cascades, stretching across ecosystems boundaries [23]. This suggests that organisms with complex cycles are one of the most common and important conduits linking the dynamics of seemingly different ecosystem types.

In this study, we use a modelling bio-inspired approach to analyse the dynamics of two digital ecosystems that are indirectly connected through agent species with a complex cycle. In particular, we examine how changes in the resources in either digital ecosystem or changes in agent affect the dynamics of the abundance and distribution of agents and resources. By accounting for the independent effects of the different resources on agents, we show that such systems can exhibit alternative stable states, which can lead to abrupt changes in the agent population structure and which can lead to complex behaviours. In general, we show that the dynamics of the system depend on the relative proportion of resources, showing the importance of indirect interactions between different digital ecosystem types.

2 Catastrophic Events in Digital Ecosystems

A large number of real systems exhibit dynamics that bear the potential for chaos. The domains over which stability of the systems occurs can be relatively large. But once in a while, systems may move towards the edge of stability and little nudges to the system may move it from stability to instability, that is, into a catastrophe. Subsequently, re-organisation of system components, which may occur as a reaction to bring the system back into a stable domain, a kind of evolutionary process. This stable domain, however, may not be the same as the one prior to the disturbance. In this new stable domain a novelty spreads by meanings, objects and subjects, and produces innovation.

The system undergoes a catastrophic event in the sense that it is moved from an initial state of stability through a dramatic phase of reorganisation and back to some degree of stability. Examples of such catastrophic events include landslides, avalanches, earth-quakes, and pest outbreaks in ecosystems. In each case, small changes in the system occur, where these individually may not be critical to the system's behaviour. Collectively, however, they lead to the evolution of the system towards a critical state. This is apparent, for example, in the case of pest outbreaks in ecosystems. Each infected individual potentially adds to the instability of the system. When a critical point is reached, the next infected individual may trigger an epidemic that affects a larger part of the system. Temporary stability is quickly reached if the epidemic is not too intense. Even if not in a large scale, the epidemic adds to the stress of the system's fragile regions, making these more susceptible to further epidemics as more individuals are infected at those regions or as additional diseases are received from the most fragile regions of the system. Ultimately, a large-scale catastrophic event may occur that affects the entire system, not just individual regions. The system components regroup and finally enter a phase of new, temporary stability.

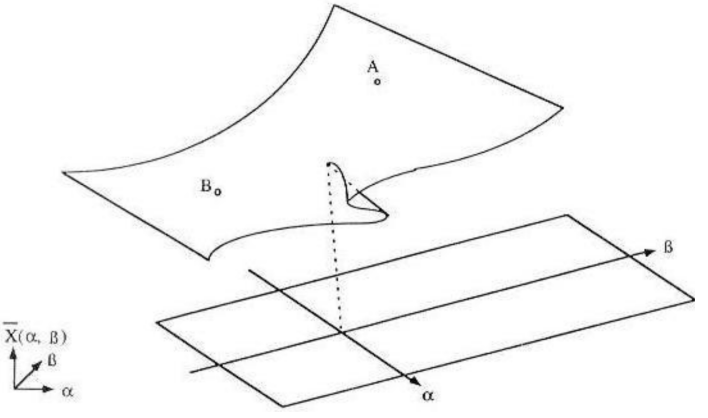


Fig. 2.

Evolutionary processes are hence at work, making the system more efficient. We refer to this as evolution toward catastrophe. A system in such a state can transgress to a stable state by another process of evolution, faster than the first kind, where this new stable state may not be very efficient. Large living natural systems are similarly constrained from operating at or near peak efficiency by random intervention of uncoordinated external processes at the regional levels.

We can develop a simple model of catastrophe and then proceed to create an epidemic model. Before this, we consider the following equation [24](Beltrami, 1998):

$$x^3 - \alpha x - \beta = 0\,. \tag{1}$$

Consider Figure 2, which illustrates the surface defined by Eq.(1). Imagine a ball lying at the top of this surface such as point A. The ball may be still, and very small nudges away from its first equilibrium point A, leading to a new equilibrium. After a series of such small perturbations, however, the ball will roll off the top part of the surface, and a priori it is difficult for us to determine exactly where it will end up. All we know for certain is that the new equilibrium position is somewhere at the bottom of the surface, say point B. Small nudges to the ball in B will again move it slightly away from B. And if we push it hard enough, we can propel the ball through the fold, or cusp, to the upper part of the surface again. Where exactly will it end up? A precise answer requires exact knowledge of the shape of the surface, the properties of the ball, and the magnitude and direction of the force exerted on the ball. In more complicated real-life systems not all the variables describing the system and the forces exerted are known well enough. As a result, we may only know stability domains rather than specific locations.

3 Budworm Dynamics

A classic example for the implications of ecosystem catastrophes is budworm dynamics [25](Royama, 1984). Budworms are caterpillars that feed on spruce and fir forest in

north-eastern Canada and the United States. When forest stands reach maturity, budworm populations explode, seriously affecting the forest by defoliating the trees. With the death of trees comes a loss of food and consequent population crash. The cycle of [low population density]-[stand maturity]-[budworm population explosion (and collapse)] tends to repeat itself over a lapse of time. A natural systems way of controlling, amongst others, relates to the patch size. Natural systems no doubt avoid large catastrophes because they operate in patches, where the degree of maturity of adjacent tree patches is nearly always different. Consequently, pests and disease find difficulty in spreading beyond a patch, and the size of the catastrophe is kept small. Current MAS practice seems to ignore such system behaviour. Our model is not a regional one and such inter-patch dynamics are not captured. However, we have conceptualised such patch dynamics to combine with this conceptual-theoretical model.

To model a bio-inspired MAS catastrophe, we use a budworm dynamics model (see [24]Beltrami, 1998). Let us denote B as the agent population density, k the information carrying capacity, S the network density, and r_B the agent population growth rate. Thus,

$$\frac{dB}{dt} = r_B B\left(1 - \frac{B}{kS}\right). \tag{2}$$

describes the population dynamics for a fixed information carrying capacity and no negative influences on agent population growth. This is a logistic growth equation. (Figure 3).

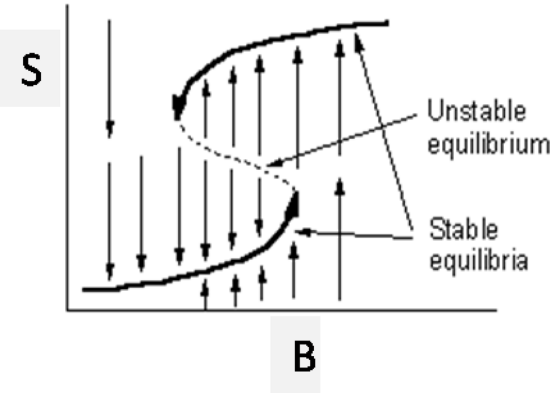


Fig. 3. The relation between network density (S) and agent population (B)

Let us introduce the effects of a negative agent population growth factor, with a maximum rate of c, which is assumed to be constant. At small densities, c has only a minor effect on agent population because it is small, and thus the probability of those affected by the negative factor is also small. A negative agent population growth factor that captures such interactions is:

$$\frac{cB^2}{a^2 + B^2}. \tag{3}$$

With *a* as a scalar that captures the effectiveness of the 'negative agent population growth factor' to affect and eradicate an agent. In a developed (more populated) system of agents, it is assumed that the negative factor is more effective than in a less developed system. Thus, *a* may be assumed to increase with increased development of the MAS, that is, network density

$$a = mS \ . \tag{4}$$

And thus

$$\frac{cB^2}{a^2 + B^2} = \frac{cB^2}{(mS)^2 + B^2} \ . \tag{5}$$

Where m is a constant. Combining negative population growth with the logistic function yields

$$\Delta B = \frac{dB}{dt} = r_B B\left(1 - \frac{B}{kS}\right) - \frac{cB^2}{(mS)^2 + B^2} \ . \tag{6}$$

Which is the equation used in the model that drives agent population changes, ΔB. Changes in network density are assumed to also follow the logistic growth curve, with r_S as the rates of increase and k_S as information carrying capacity.

$$\text{x + y = z } \Delta S = \frac{dS}{dt} = r_S S\left(1 - \left(\frac{S}{k_S E}\right)\right) . \tag{7}$$

Where *E* is the percentage of nodes of the network. The denser the network, the higher *E* becomes. The percentage of nodes is assumed to decrease as the average agent density per network density *B/S* increases. To model the reduction of stress as agent population decreases, we multiply B/S by E^2. The combined effect of logistic growth in network and agent induced nodes loss is:

$$\Delta E = \frac{dE}{dt} = r_E E(1 - E) - \frac{pBE^2}{S} \ . \tag{8}$$

Where r_E is the rate of nodes increase and *p* a proportionality factor. Let us consider the case of $B \neq 0$ and introduce the following notation:

$$R = \frac{r_B mS}{c} \ . \tag{9}$$

$$Q = \frac{k}{m} \ . \tag{10}$$

And rewrite

$$B = mSx \ . \tag{11}$$

It can be shown that the nontrivial equilibrium of Eq. (6) fulfil

$$R\left(1-\frac{x}{Q}\right) = f(x) = g(x) \ . \tag{12}$$

With

$$g(x) = \frac{x}{1+x^2} \ . \tag{13}$$

The left side of Eq. (12) is a straight line *f(x)* with slope *–R/Q*. Equilibrium occurs where this line intersects with *g(x)*. Both, *S* and *R* increase in *Q*. At first, there is a single equilibrium, corresponding to the situation shown in Figure 1.

Here, we have studied that resource-dependent traits and an environment switch in ONE that characterize the agent-agent interactions can couple the dynamics of otherwise independent ecosystems and lead to surprisingly complex dynamics, including alternative stable states. As a consequence, bottom–up or top–down effects in either ecosystem can lead to counterintuitive shifts in the abundances of the respective agent stages and lead to dramatic regime shifts across different ecosystems, complex cycles and regime shifts across ecosystems There is increasing evidence of dramatic regime shifts in different ecosystems in nature, and it has been suggested that this is due to a switch between different alternative stable states (ASS) of the system [27], [28], [29]. Here, we have studied for the ONE that enrichment in either resource or increasing the agent stage can lead to a shift between ASS in different ecosystems simultaneously.

Once a critical threshold is reached, the agent population shows a dramatic shift from a mature to an immature dominated population or vice versa, which results in equally dramatic but opposite shifts in their respective resources. Surprisingly, this leads to a counterintuitive collapse of the agent stage of the enriched resource whereas the other stage dramatically increases. In general, these approaches suggest that agent with complex cycles can create alternative stables states that can lead to dramatic regime shifts across completely different open negotiation environments in digital ecosystems.

Coupling dynamics of natural ecosystem through complex cycles and fluxes of species with complex cycles are a major and important conduit connecting different ecosystems [30], [31], [32]. Little is known about the effects of the reciprocal interaction between both ecosystems that necessarily result from agents with complex cycles. Here, we have studied that changes in one ecosystem can indirectly and dramatically alter the community structure in the connected ecosystem if they are connected through agents with complex cycles. Previous studies in nature have shown that there is substantial variation in the relative difference between mature and immature capacities across and within different species and ecosystems [33].

However, decreasing the proportion of the mature environment can bring about a shift between ASS if the agent population is initially dominated by immature agents.

These approaches suggest that the indirect interactions between ecosystems through agents with complex cycles are often not symmetrical and that we need to account for the distribution of the agent negotiation styles to predict how behavioural changes will affect both environments.

4 Final Remarks

Although there is evidence of complex shifts of agents between environments and ecosystems, there is surprisingly little information about these shifts for the long-term dynamics of coupled ecosystems, most likely due to the difficulties to monitor the different stages and styles of agents with complex cycles and their respective resource use. Our approach emphasize how reciprocal agent fluxes across utterly different environment and ecosystems can affect the structure and dynamics of local agent interactions and emphasize the need to account for the resulting indirect interaction between them. These approach emphasize the need to think across ecosystems when studying community dynamics and managing agent populations with complex cycles. A further body of theory treats ecosystems as complex adaptive systems [34]. These models provide a theoretical basis for the occurrence of self-organization in both digital and real ecosystems. Self-organization results when interactions among agents and their environment giving rise to complex nonlinear behaviour. It is this property that provides the underlying potential for scalable problem-solving in a digital environment.

By comparing and contrasting theoretical ecology with the anticipated requirements of digital ecosystems, we have examined how ecological features may emerge in some systems designed for adaptive problem solving. Specifically, we suggested that a digital ecosystem, like a real ecosystem, will usually consist of self-organized agents that interact both with one another and with an environment. Agent population dynamics and evolution, spatial and network interactions, and complex dynamic fitness landscapes will all influence the behaviour of these systems. Many of these properties can be understood via well-known ecological models [35], [36].

Acknowledgements. This work was supported in part by the grant Nº 34744 (ONE) – Open Negotiation Environment, FP6-2005-IST-5, from the European Union, by the Generalitat de Catalunya through Grant SGR-00296, by the FI-AGAUR Research Fellowship Program, Generalitat de Catalunya, Contract 2006-FI-00230, and by the European Social Fund of the European Union, ESF-EU.

References

1. Ximbiotix, About the digital ecosystem (2006), http://www.ximbiotix.com/desktop/the-digital-ecosystem/about-the-digital-ecosystem.cfm
2. Fiorina, C.: The digital ecosystem, Hewlett Packard (2006), http://www.hp.com/hpinfo/execteam/speeches/fiorina/ceo_worldres_00.html
3. Syntel, Building your own digital ecosystem: a holistic approach to enterprise integration (2006), http://www.syntelinc.com/uploadedFiles/SyntelDigitalEcosystem.pdf

4. Silicom, Digital ecosystem (2006), http://www.silicom.com/de1.shtml
5. Vandenberghe, M.: The business factory - digital ecosystem solution (2006), http://themaddesigner.free.fr/XeWOW%20White%20Paper.pdf
6. Yaeger, L.: Computational genetics, physiology, metabolism, neural systems, learning, vision and behavior or polyworld: Life in a new context, in Artificial Life III, SFI Studies in the Sciences of Complexity, Langton, pp. 263–298. Addison-Wesley, Reading (1993)
7. Ray, T.S.: An approach to the synthesis of life. Artificial Life II. Santa Fe Institute Studies in the Sciences of Complexity, vol. XI, pp. 371–408. Addison-Wesley, Redwood City (1991)
8. Adami, C., Brown, C.T.: Evolutionary learning in the artificial life system "avida". In: R.B., Maes, P. (eds.) Artificial Life IV: Fourth International Workshop on the Synthesis and Simulation of Living Systems, pp. 377–381. MIT Press, Cambridge (1994)
9. Pargellis, A.N.: Digital life behavior in the amoeba world. Artificial Life 7, 63–75 (2001)
10. Ofria, C., Wilke, C.O.: Avida: a software platform for research in computational evolutionary biology. Artificial Life 10, 191–229 (2004)
11. Digital Business Ecosystems project, EU Framework 6 Integrated Project (2006), http://www.digital-ecosystem.org/
12. Callebaut, W., Rasskin-Gutman, D. (eds.): Modularity. Understanding the Development and Evolution of Natural Complex Systems. MIT Press, Cambridge (2005)
13. Gallagher, S., Varela, F.J.: Redrawing the map and resetting the time: Phenomenology and the cognitive sciences. Can. J. Phil. 29, 93–132 (2001)
14. Varela, F.J., Thompson, E., Rosch, E.: The Embodied Mind: Cognitive Science and Human Experience. MIT Press, Cambridge (1991)
15. Noë, A.: Action in Perception. MIT Press, Cambridge (2005)
16. Noë, A.: Experience without the head. In: Perceptual Experience. Edited by Tamar Szabo Gendler and John Hawthorne, OUP (2006)
17. Polis, G.A., Anderson, W.B., Holt, R.D.: Toward an integration of landscape and food web ecology: the dynamics of spatially subsidized food webs. Annu. Rev. Ecol. Syst. 28, 289–316 (1997)
18. Polis, G.A., Power, M.E., Huxel, G.R.: Food webs at the landscape level. University of Chicago Press, Chicago (2004)
19. Werner, E.E.: Size, scaling and the evolution of complex life cycles. In: Size-structured Populations, pp. 60–81. Springer, Berlin (1988)
20. Wilbur, H.M.: Complex life-cycles. Annu. Rev. Ecol. Syst. 11, 67–93 (1980)
21. Werner, E.E., Gilliam, J.F.: The ontogenetic niche and species interactions in size structured populations. Annu. Rev. Ecol. Syst. 15, 393–425 (1984)
22. Polis, G.A., Meyers, C.A., Holt, R.D.: The ecology and evolution of intraguild predation: potential competitors that eat each other. Annu. Rev. Ecol. Syst. 20, 297–330 (1989)
23. Knight, T.M., McCoy, M.W., Chase, J.M., McCoy, K.A., Holt, R.D.: Trophic cascades across ecosystems. Nature 437, 880–883 (2005)
24. Beltrami, E.: Mathematics for Dynamic Modeling, 2nd edn. Academic Press, London (1998)
25. Royama, T.: Population dynamics of the spruce budworm Choristoneura fumiferana. Ecological Monographs 54(4), 429–462 (1984)
26. Scheffer, M., Carpenter, S., Foley, J.A., Folke, C., Walker, B.: Catastrophic shifts in ecosystems. Nature 413, 591–596 (2001)
27. Scheffer, M., Carpenter, S.R.: Catastrophic regime shifts in ecosystems: linking theory to observation. Trends Ecol. Evol. 18, 648–656 (2003)

28. Schröder, A., Persson, L., De Roos, A.M.: Direct experimental evidence for alternative stable states: a review. Oikos 110, 3–19 (2005)
29. Polis, G.A., Holt, R.D., Menge, B.A., Winemiller, K.O.: Time, space, and life history: influences on food webs. In: Polis, G.E., Power, M.E., Huxel, G.R. (eds.) Food Webs: Integration of Patterns and Dynamics, pp. 435–460. Chapman & Hall, New York (1996)
30. Nakano, S., Murakami, M.: Reciprocal subsidies: Dynamic interdependence between terrestrial and aquatic food webs. Proc. Natl. Acad. Sci. U.S.A 98, 166–170 (2001)
31. Marczak, L.B., Thompson, R.M., Richardson, J.S.: Meta-analysis: trophic level, environment, and productivity shape the food web effects of resource subsidies. Ecology 88, 140–148 (2007)
32. MacArthur, R., Wilson, E.O.: The Theory of Island Biogeography. Princeton University Press, Princeton (1967)
33. Hubbell, S.: The Unified Neutral Theory of Biodiversity and Biogeography. Princeton Univ. Press, Princeton (2001)

The User's Human Values Scale Methodology in Recommender Systems from Several Information Sources of the Organization

Javier Guzmán-Obando[1,2], Josep Lluís de la Rosa[1], Silvana Aciar[1], Miquel Montaner[1], José A. Castán[2], and Julio Laria[2]

[1] University of Girona, Campus Montilivi,
Ave. Lluís Santaló, Building P-IV, Girona, Spain
{jguzmano,peplluis,saciar,mmontaner}@eia.udg.edu
[2] Universidad Autónoma de Tamaulipas,
Facultad de Ingeniería "Arturo Narro Siller",
Tampico, Tams. México
jguzmano@uat.edu.mx

Abstract. The main objective of this paper aims at developing a methodology that takes into account the human factor extracted from the several information sources of the organization of the recommender systems. This methodology is capable of extracting the Human Values Scale from the user, with reference to his/her features, in order to improve the adaptation of the Recommender Systems. This research is focused on the analysis of human values scale using the Portrait Values Questionnaire of Schwartz, which can take advantage of the several information sources of the organization through its attributes to define the methodology that response with more exactitude to preferences and interests of the user. This paper presents a demonstration of how the Human Values Scale of a user can be extracted from several information sources of the organization. A case study is presented to apply the methodology, in an effort to extract the user human values scale from bank domains.

Keywords: Human Values Scale, information sources, Recommender Systems.

1 Introduction

Personalization of services using a user's Human Values Scale (HVS) can improve user satisfaction. According to (Jensen, 2002), the information society will be followed by a society in which individuals will prioritize their decisions in interactions that involve a high degree of emotion, which will be a relevant issue in their values scale. Therefore, we are witnessing a cyclical transformation in society affecting its values scales.

In traditional psychology (Schwartz, 2003), the HVS defines a set of desirable and nonsituational goals; their significance can vary from one person to another and govern their life like a set of individual principles.

Organizational recommender systems represent user preferences for the purpose of suggesting items to purchase or examine. They have become fundamental applications

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 900–912, 2008.

in electronic commerce and information access, providing suggestions that effectively prune large information spaces so that users are directed toward those items that best meet their needs and preferences (Burke, 2002).

However, in the next stage of recommender systems, users will be situational humans who make decisions based not only on their preferences, tastes and interests, but also on their perceptions about them. In recommender systems, emotional sensibility can be defined as the emotional response of the user to the suggestions, advice or predictions of interest made by the system in each particular context, obtained through the objective, subjective and emotional attributes of the Several information sources of the organization (Gonzalez, et.al., 2005).

Our research is focused on the analysis of HVS using the Portrait Values Questionnaire (PVQ) (Schwartz, 2003), which can take advantage of the information sources of the organization through its objective, subjective and emotional attributes to define a methodology that responds more exactly to the preferences and interests of the user. This kind of methodology can influence the user's perception and final decision making.

The paper is organized as follows. In Section 2, a brief introduction to work related to the HVS is presented. Section 3 gives a study of the HVS in several information sources of the organization to understand the emotional response in recommender systems. In Section 4, we describe the methodology proposed to obtain the HVS of the user from information sources of the organization. Next, in Section 5, we illustrate the proposed methodology through a recommender system of banking services. Finally, Section 6 offers some conclusions and suggestions for future work.

2 Related Work

Research studies (Ravlin and Meglino, 1987) have proved the influence of human values on the perception and decision making of human beings. These studies reveal the value structure of each individual, in particular the values to which a greater or smaller importance is assigned, as they play a determining a role in perception as in decision making. We carried out an analysis of the most used scales on research for measuring human values (Guzman, et.al, 2005). Some do not measure the range of human values relevant in many life domains; others, despite their intention of covering the range of human values comprehensively, leave out critical content (e.g., tradition and power values); in other cases some items are highly sensitive to prevailing economic conditions and measure individuals' values only indirectly. We consider that the most suitable technique to apply in this research is the Schwartz scale of values, as it covers 56 human values included in 10 basic values. The reliability and validity of the Schwartz value survey have been demonstrated in several studies (Gouveia, et.al., 1998) and (Schwartz, 2003).

The Schwartz Value Survey (Schwartz, 1999) consists of 40 items, each one associated with an asymmetric scale from one (opposed to personal values) to six (of supreme importance), indicating the importance of this value as a guiding principle in the user's life.

The survey items are distributed among 10 universal dimensions (see Figure 4.a), which respond to different underlying motivations of the values integrating them. We

call these dimensions meta-attributes. They are grouped taking into account compatible typologies and the diametrically opposed incompatible typologies (see Figure 1), which represent a contradiction of objectives that would generate a conflict in the user.

The procedure for scoring agreement to the PVQ is as follows:

1. apply the PVQ;
2. to obtain the personal score in a typology, add the points that have been assigned to questions associated with that typology;
3. divide the result by the number of questions associated with the typology;
4. mark thescore of each typology in the corresponding axis of the Dynamic Structure of Values; and,
5. connect the points until a polygon of 10 sides is completed.

This procedure allows the HVS of a user to be developed from existing several information sources of the organization (Guzman, et.al., 2005).

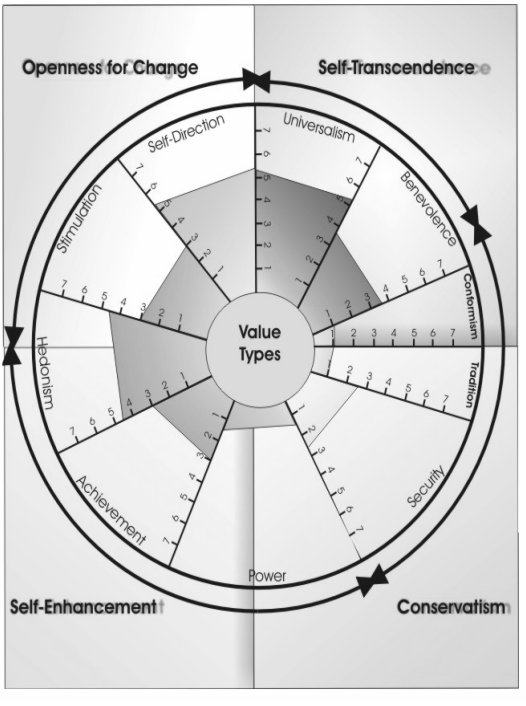


Fig. 1. Dynamic Structure of Values according to the Universal Theory of Schwartz

3 The Human Values Scale in Several Information Sources of the Organization for Recommender Systems

The information sources of the organization represents assumptions about the user's knowledge, beliefs, preferences, and other user characteristics (Kobsa, 2001).

The values scale in information sources can be defined as the set of rules that manage the behaviour of a flexible autonomous entity, which is related with the attributes of the user (Guzman, et.al., 2005).

In our research, the general information about a user is useful for a recommendation process, because one can deduce that the values scale can be applied to autonomous and flexible entities, for instance a multiagent (Gonzalez, et.al., 2005), for the following reasons.

- To measure the interests and preferences of a social entity is useful.
- It motivates actions and gives them direction and emotional intensity.
- It functions as a criterion scale to evaluate and justify the actions.
- It is acquired both through the experience of individual learning and through the socialization in the values of a group of socially intelligent agents.

Values act as a central means of rationalizing actions within the human mind. Given a goal, values dictate the way in which the goal will be accomplished (Carter and Ghorbani, 2004).

The values scale is represented by goals (implicit or explicit) that reflect the needs of every flexible and autonomous social entity so the scale can:

- establish social relationships and coordinate them;
- express goals, objectives and interests explicitly; and
- create clusters with similar characteristics and social interests.

The HVS is an integral approach to information sources and can take advantage of the information sources of the organization using its attributes to use them in the processes of the recommendation (Guzman, 2005).

4 Obtaining the HVS of a User

To calculate the HVS of a user, we must first obtain the user's general characteristics from the information sources of the organization by applying the PVQ. Then, through the proposed method, support will be given to the recommender system to make suggestions as a function of the user HVS.

4.1 The HVS Test

The PVQ provides a set of universal values classified into four groups: Openness to Change, Conservation, Self-transcendence, and Self-enhancement. These four groups are further divided into 10 sets of universal values (described above), which have n items to calculate the user HVS. Each item is defined in [0, 1]. (see Figures 2 and 4.a).

4.2 Information Distribution

The values of the attributes from the recommender system provide relevant information about the user, from which we wish to obtain the HVS.

Meta-Attribute	Qualification (SVS)	Normalize Qualification
Self-Direction	3	0.50000
Benevolence	3.5	0.58333
Conformity	2	0.33333
Stimulation	1	0.16667
Hedonism	1	0.16667
Achievement	4	0.66667
Power	3	0.50000
Security	5	0.83333
Tradition	6	1.00000
Universalism	6	1.00000

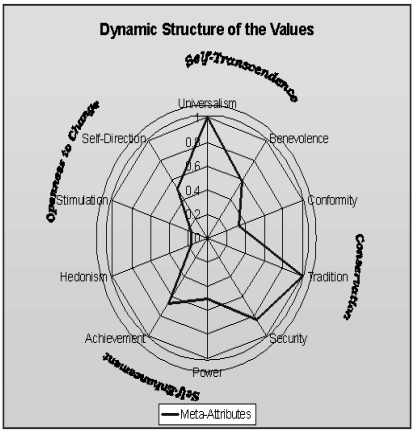


Fig. 2. Example of the results from a PVQ

In our model, the technique represents the values as points in a multidimensional space. Distances between points reflect empirical relations between the values that can be measured by the correlations between the scores that give their importance for the person. A bigger conceptual similarity between two values shows that they are more related empirically, and therefore they will be closer in the multidimensional space. Figure 2 shows the items related to the HVS.

According to Figure 4, the set of parameters that define the HVS are:

$$Evh = \{Vu_1, ..., Vu_n\} \tag{1}$$

where the *Vu* are the universal values such as openness to change, conservatism, self-transcendence and self-enhancement.

$$Vu = \{Vh_1, ..., Vh_n\} \tag{2}$$

The *Vh* are the human values corresponding to the 10 types described by Schwartz: universalism, benevolence, conformism, tradition, security, achievements, power, hedonism, self-direction and stimulation.

$$Vh = \{a_1 ..., a_n\} \tag{3}$$

The *a* values correspond to the attributes or particular items, such as equality, intelligence, social order, richness, or creativity. In this way, we have: $\forall a_i \in Vh$ have a $val(a_i) \in [0,1]$; $\forall Vh \in Vu$ with $val(v_i) \in [0,1]$; and, $\forall Vu \in Evh$

At the end, each a_i in *Vh* has a value. Once the corresponding values are obtained, the user HVS is generated from the information sources of the organization with val(u_i) in [0,1].

4.2.1 Recommendation

The objective of this stage is to take advantage of the user's HVS to provide information to the organizational recommender system to improve the recommendations made to the user. To achieve this objective, the following methodology will be used.

4.2.2 Calculation of Numerical Values

In this stage, the calculations are made to obtain the user HVS, following a series of steps.

1. Obtain the individual qualification of each *Vh*.
2. Calculate the numerical value of each *Vu*.
3. Perform the global calculation of the user HVS.

Step 1: In this step the value of each *Vh* is obtained by composing the user HVS. For each *Vh* there is a set of values (attributes, items) given by:

$$\forall q \in Q, \exists Vh(q_i) \subset Vu \mid$$

$$Val\_num(q_i) = \frac{\sum_{j=1}^{j=n_a} Val\_num(a_j)}{n_a} \in [0,1] \tag{4}$$

Where: n_a = number of attributes evaluated in *Vh*.

Step 2: In this step the qualification of each *Vu* is calculated for the user HVS. For each *Vu* there is a set of universal values given by:

$$\forall q \in Q, \exists Vu(q_i) \subset Evh$$

$$Val\_Num(q_i) = \frac{\sum_{i=1}^{i=n_{Vh}} Val\_Num(Vh_i)}{n_{Vh}} \in [0,1] \tag{5}$$

Where: n_{Vh} = number of type values evaluated in *Vu*.

Step 3: In this last step, calculate the final value that corresponds to the user HVS, as follows:

$$Evh = \frac{\sum_{j=1}^{j=nVu} Val\_Num(Vu_j)}{n_{Vu}} \in [0,1] \tag{6}$$

Where: n_{Vu} = total number of universal values in the HVS.

4.2.3 Linguistic Labels

Traditionally, modifications of fuzzy sets called linguistic labels, equivalent to adverbs, have been used. The interpretation in the fuzzy model of these enunciates involves assigning a value to the belong function with a simple arithmetic calculation.

According to the PVQ, the answers to the survey items range from it is not like me to it is very much like me. In this case, we represent the facts defining each set so that each element belongs to the set with a certain degree (possibility).

In a more formal way, a fuzzy set A is characterized by a belong function: μA : U → [0,1] that associates to each element *x* of *U* a number μ*A(x)* from the range [0, 1] that represents the degree of belonging of *x* to the fuzzy set *A*. *U* is called the universe of speech. The fuzzy terms for the example studied can be defined by a trapezoidal fuzzy set:

$$\mu\tilde{A}(x) = \begin{cases} 0 & ; \quad x \le a_1 \\ \dfrac{x - a_1}{a_2 - a_1} & ; \quad a_1 \le x \le a_2 \\ \dfrac{a_3 - x}{a_3 - a_2} & ; \quad a_2 \le x \le a_3 \\ 0 & ; \quad x \ge a_4 \end{cases}$$

In this way, we obtain a graph representing the linguistic variable x by fuzzy logic, as shown in Figure 3.

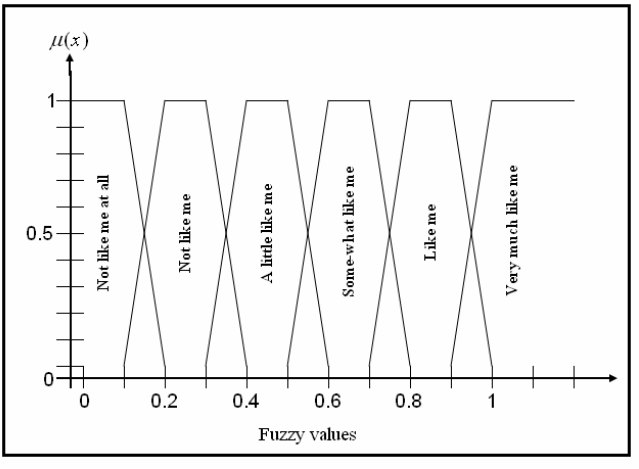


Fig. 3. Functions [0,1]

4.2.4 Generating Recommendations

Although the effectiveness of existing recommender systems is remarkable, they still have serious limitations as they are unable to perform qualitative inference on the recommendations they offer and are incapable of dealing with the defeasible nature of users' preferences. In this context, defeasible argumentation frameworks (Chesñevar, et.al., 2003) have evolved to become a sound setting to formalize commonsense qualitative reasoning.

In recent years several specialized techniques for improving Web searches have been developed. Most existing approaches are still limited, mainly because of the absence of qualitative criteria for ranking results and insensitivity to user preferences for guiding the search.

At the same time, defeasible argumentation evolved as a successful approach in AI to model commonsense qualitative reasoning with applications in many areas, such as agent theory, knowledge engineering and legal reasoning (Chesñevar and Maguitman, 2004).

We contend that defeasible argumentation, the methodology proposed in (Chesñevar and Maguitman, 2004), can be integrated into existing recommender system technologies, paving the way to solve the problems of the recommender processes. We will analyse our proposal in the next section, illustrating it with a case study.

5 Illustration

We illustrate the proposed methodology through a recommender system for banking services.

Nowadays, banks offer their customersproducts and services using recommender systems, taking into account their interests, preferences and attitudes, and the user's interactions with the system (transactions).

Several information sources of the organization registers the user's movements so that the recommender system can offer more suitable solutions that will increase the confidence of the customer in the banking organization. This allows the bank to know the customer, interpreting his or her necessities, capacities and attitudes to consumption.

Banking transactions that would help the recommendation process include card contracts, relationship indicators, movements of current account, domiciled invoices, card movements, and income.

5.1 Description

Assume an intelligent agent A with a knowledge base *(K, Δ)* that contains information on the conditions under which Services Bank A, according to the HVS of their customers, offer a credit service to purchase a high technology hybrid vehicle.

To offer credit for this t ype of product, the bank conditions are that the customer:

1. must have a high degree of openness to change;
2. must have a high score in the values group of self-enhancement;
3. because of the product/service offered, the level of conservatism must be less that 50%; and,
4. the level of self-transcendence must be less than 50%.

In this study, we make the analysis using attributes from the customer Jordi Vilà. We refer to Figure 4.b, in which we represent the values for each of the items extracted from the information sources of the organization to obtain the corresponding calculations.

5.2 Applying the Methodology

1. Following the methodology proposed by (Guzman, 2005), the general characteristics of the user are obtained through the information sources of the organization that computes the user data for the bank's recommender system.
2. According to (3), and as a result of applying the PVQ, we obtain the following results.

$$Vh = \begin{Bmatrix} val(broad\ \min ded, 0.60), val(Wisdom, 0.40), \\ val(social\_justice, 0.40), .., (an\_exciting\_life, 0.80) \end{Bmatrix}$$

We calculate the users HVS from the information sources of the organization.

$$Val\_Num(Universalism) =$$

$$= \frac{\begin{array}{l} val(broad\ \min ded) + val(wisdom) + val(social\_justice) + val(equality) + \\ val(a\_world\_at\_peace) + val(unity\_with\_natura) \end{array}}{6}$$

$$= \frac{2.40}{6} = 0.40$$

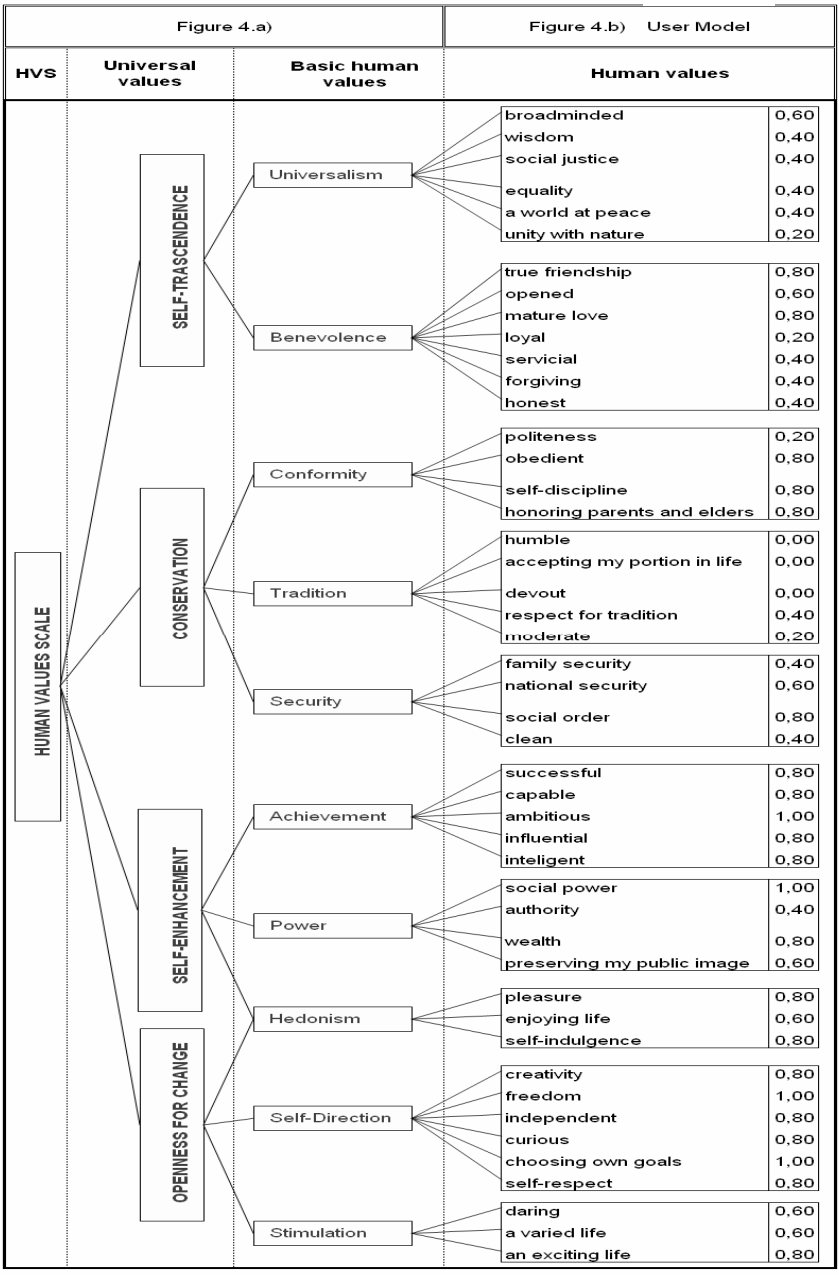


Fig. 4. Parameters tree to classify the HVS

Step 1: Applying equation 4, we obtain the 10 human values of the user as follows.

In the same way we calculate the other human values:

$$Val\_Num(Benevolence) = 0.51 ; Val\_Num(Conformity) = 0.65 ;$$
$$Val\_Num(Tradition) = 0.12 ; \ Val\_Num(Security) = 0.55 ; \ Val\_Num(Achievement) = 0.84 ;$$
$$Val\_Num(Power) = 0.70 ; \ Val\_Num(Hedonism) = 0.73 ; \ Val\_Num(Self\_direction) = 0.87 ;$$
$$Val\_Num(Stimulation) = 0.67$$

Step 2: Using equation 5, we calculate the 4 groups which correspond to the universal values of the HVS

$$Val\_Num(Self\_trascendence) = \frac{Val\_Num(Universalism) + Val\_Num(Benevolence)}{2}$$
$$= \frac{0.91}{2} = 0.46$$

Analogously we can compute the next 3 universal values, obtaining:

$$Val\_Num(Conservation) = 0.44 ;$$
$$Val\_Num(Self\_Enhancement) = 0.76 ;$$
$$Val\_Num(Openness\_to\_change) = 0.76$$

Step 3: In this last step we calculate the user HVS using equation 6.

$$Evh = \frac{Val\_Num(Self\_trascendence) + Val\_Num(Conservation) + Val\_Num(Self\_enhancement) + Val\_Num(Openness\_to\_change)}{4}$$
$$= \frac{0.46 + 0.44 + 0.76 + 0.76}{4} = \frac{2.42}{4} = 0.60$$

Once obtained the HVS we can carry on with the recommendation process by using an approach with defeasible argumentation.

5.3 Alternative with Defeasible Argumentation

One solution to the problem is by using defeasible argumentation. Considering that the bank conditions for offering the credit service to Jordi Vilà affect the four main universal values groups directly, these will be considered in the recommender process.

5.3.1 First User Case

If (K, Δ) is the knowledge base defined in 5.1 The group PredName(K, Δ) associated can be divided in the following way:

$$\Pr edBase(L) = \{conservation, self\_trascendence\} ; \text{ and,}$$
$$\Pr edDeriv(L) = \{openness\_to\_change, self\_enhancement, offer\_service\}$$

Therefore, the logic defeasible argumentation is as follows:

$$\Delta = \left\{ \begin{array}{l} Openness\_to\_change(x) \wedge Self\_enhancement(x) \\ \succ offer\_service(x, y), \\ Openness\_to\_change(x) \succ \neg Conservation(x), \\ Self\_enhancement(x) \succ \neg Self\_trascendence(x) \end{array} \right\}$$

$$\kappa_G = \left\{ \begin{array}{l} Conservation(x) \rightarrow \neg Openness\_to\_change(x), \\ Self\_trascendence(x) \rightarrow \neg Self\_enhancement(x) \end{array} \right\}$$

Considering the universal values group from Jordi Vilà, we have:

The possible values to measure the grade or level of score from the universal values is given by:

$$L_n : nvalores\{0,1/(n-1),2/(n-1),...,(n-2)/(n-1),1\}$$

Therefore:

$$A(x) = very\_high(A_i(x)) = A_i(x) = 1$$
$$A(x) = high(A_i(x)) = A_i^{0.75}(x)$$
$$A(x) = medium(A_i(x)) = A_i^{0.50}(x)$$
$$A(x) = low(A_i(x)) = A_i^{0.25}(x)$$
$$A(x) = very\_low(A_i(x)) = A_i^{0}(x) = 0$$

Then,

$$Openness\_to\_change(Jordi\_Vila)) = 0.76 > 0.75$$
$$Self\_Enhancement(Jordi\_Vila) = 0.76 > 0.75$$
$$Conservation(Jordi\_Vila) = 0.44 < 0.50$$
$$Self\_trascendence(Jordi\_Vila) = 0.46 < 0.50$$

Note that in all four groups, conditions for the bank to offer the service to the user Jordi Vilà are fulfilled. Therefore the system recommends sending the brochure and all the details regarding the credit service to this distinguished customer.

5.3.2 Second User Case

Suppose that for the same problem, we have the HVS from the user Montse Rovira, which data are the following:

$$Openness\_to\_change(Montse\_Rovira)) = 0.77 > 0.75;$$
$$Self\_Enhancement(Montse\_Rovira) = 0.75 = 0.75;$$
$$Conservation(Montse\_Rovira) = 0.52 > 0.50;$$
$$Self\_trascendence(Montse\_Rovira) = 0.48 < 0.50$$

In this case, according to the logic defeasible argumentation, the argument:

$$A_1 = \{Self\_trascendence(x) \rightarrow \neg Self\_enhancement(x)\}$$

Contraargument to:

$$A_1 = \{Openness\_to\_change(x) \wedge Self\_enhancement(x) \succ offer\_service(x, y)\}$$

therefore the system recommend to the bank not to offer the credit service to the user Montse Rovira.

6 Conclusions and Future Work

In this paper, we show a methodology based on defeasible argumentation to organizational recommender systems based on the user's HVS. This HVS is obtained directly from the severals information sources of the organization.

Through the proposed methodology, it is possible to calculate the HVS from the information sources without disturbing the user with surveys.

The preliminary results obtained from the illustration, in section 5, on banking services show that the HVS of the users is important to improve the recommender processes in the organization.

In future, we will work on the implementation of this methodology to test it using machine-learning techniques to obtain better recommendations in multiple areas. It is intended to extend the use of defeasible argumentation to obtain diverse techniques to improve the user's decision making.

Acknowledgements

This work was supported through Autonomous University of Tamaulipas; in part, by the Grant TIN2006-15111/Estudio sobre Arquitecturas de Cooperación from the Spanish government and by EU project Nž 34744 ONE: Open Negotiation Environment, FP6-2005-IST-5, ICT-for Networked Businesses.

References

1. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12(4) (2002)
2. Chesñevar, C., Maguitman, A.: ARGUENET: An Argument-Based Recommender System for Solving Web Search Queries. In: IEEE Conference on Intelligent Systems, Varna, Bulgaria (2004)
3. Chesñevar, C.I., Dix, J., Stolzenburg, F., Simari, G.R.: Relating Defeasible and Normal Logic Programming through Transformation Properties. Theoretical Computer Science 290(1) (2003)
4. González, G., López, B., de la Rosa, J.L.: A Multi-agent Smart User Model for Cross-domain Recommender Systems. In: Proceedings of Beyond Personalization 2005: The Next Stage of Recommender Systems Research, International Conference on Intelligent User Interfaces IUI 2005, San Diego, California, USA (2005)
5. Gouveia, V.V., Clemente, M., Vidal, M.A.: El cuestionario de valores de Schwartz (CVS): propuesta de adaptación en el formato de respuesta. Revista de Psicología Social 3(15) (1998)
6. Guzmán, J.: Research Work Report: Methodology to obtain the user's Human Values Scale from Smart User Models, Department of Electronics, Computer Science and Automatic Control University of Girona, Spain (2005), `http://eia.udg.es/~jguzmano/PublicacionesJGO.htm/ResearchWork-JGO`

7. Guzmán, J., González, G., de la Rosa, J.L., Castán, J.A.: Modelling the Human Values Scale in Recommender Systems: A First Approach. Frontiers in Artificial Intelligence and Applications Series Book, vol. 131. IOS Press, Amsterdam (2005)
8. Carter, J., Ghorbani, A.: Value Centric Trust in Multiagent Systems, Faculty of Computer Science. University of New Brunswick Fredericton, NB, E3B 5A3, Canada (2004)
9. Jensen, T.: New Consumers and New Communities in Consumption (2002)
10. Kobsa, A.: Generic User Modeling Systems. In: User Modelling and User-Adapted Interaction. Kluwer Academic Publishers, Netherlands (2001)
11. Ravlin, E.C., Meglino, B.M.: Effect of values on perception and decision making: A study of alternative work values Measure. Journal of Applied Psychology (1987)
12. Schwartz, S.H.: A theory of cultural values and some implications for work. Applied Psychology: An International Review (1999)
13. Schwartz, S.H.: A Proposal for Measuring Value Orientations across Nations. The Hebrew University of Jerusalem, Jerusalem (2003)

ANT Based Routing Algorithm over a HAP Network with Integrated Services

Floriano De Rango[1], Mauro Tropea[1], Apollonia Provato[2], Amilcare-Francesco Santamaria[1], and Salvatore Marano[1]

via P. Bucci 42/c 87036 Rende, Cosenza, Italy
Tel.: +39-0984-494660
Fax: +39-0984-494713
{derango,mtropea,santamaria,marano}@deis.unical.it,
apollonia1@tiscali.it

Abstract. In this paper we propose a new method of performing equally distributed routing algorithm over a network composed of a mesh of High Altitude Platforms (HAPs), based on Swarm Intelligence studies. In particular, this proposed algorithm inherits some behavior of AntNet routing but with the extensions of novel metrics for the multi-objective optimization. In order to build an optimal solution, the proposed an algorithm will make use of ANT agents that consist of probe packets sent on the HAPs network that allow to find the optimization problem solution. In this work we perform a comparison of a classical Shortest Path Algorithm and our algorithm that will try to find the minimum hop path respecting a maximum end-to-end delay bound and an equally distribution of the traffic on the HAPs network.

1 Introduction

In the last years, the HAPs [1], [2] are becoming increasingly important for global communications because capable of providing telecommunication services in rural, remote and impervious areas.

Even with today's ever-increasing computing power, there are still many types of problems (optimization problem, like path length or hop number minimization, maximum end-to-end delay minimization etc.) that are very difficult to solve. Many distributed problems have been solved by self-organizing autonomous agent systems. In such systems agents with a local view of their environment take actions that advance global system objectives. This approach is called Swarm Intelligence (SI) [3], [4], [5].

In this paper a novel probabilistic routing algorithm based on SI is proposed. This is capable to find minimum-length path satisfying specific bandwidth requirements and able to distribute traffic load over the network.

The proposed algorithm inherits some behavior of AntNet routing [6] but with the extensions of novel metrics for the multi-objective optimization [7]. In order to build an optimal solution, this algorithm will make use of ANT agents that consist of probe packets sent on the HAPs network [1]. These ANTs are useful for collecting local info

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 913–924, 2008.

for the specific problem. When an ANT goes through a node it will determine a link selection probability in order to define the optimal path towards the destination on the basis of a local heuristic and of the global info of the previous ANTs that passed in the network. The main principle behind these interactions is called *stigmergy* [3], [6], that describes indirect communication among individuals through modifications induced in their environment. The ANTs were able to find shortest paths using only the pheromone [7] trail deposited by other ANTs. The considered scenario is a HAPs mesh network because this technology presents dynamic characteristics, e.g., in the case of HAP replacement for refuel or pilot changing (see Fig.1.). These situations create the context on which to apply the SI mechanism and permit of exploiting overall characteristics of this distribute paradigm.

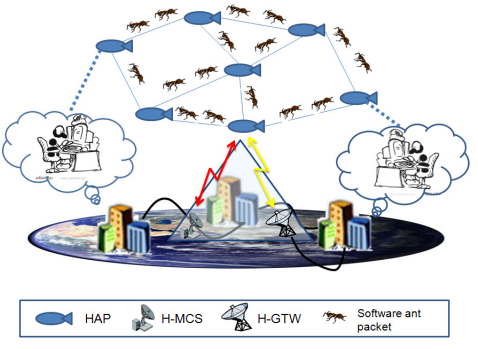


Fig. 1. Reference Scenario

In this work we have performed a comparison between the proposed algorithm with a Shortest Path Algorithm (SPA) [8] in order to test our algorithm in the search of minimum hop count path. In future works we will also analyze a comparison between our algorithms with others capable of performing QoS requirements.

Moreover, we have considered three ANT routing algorithms, an ANT algorithm with only the Minimum Hop Count metric (A_MHC), an ANT algorithm with Minimum Hop Count metric together a Maximum end-to-end Delay (A_MHC_MD) and an ANT algorithm with Minimum Hop Count, Maximum end-to-end Delay and Load Balancing metrics (A_LB). The three variants of our algorithm show how each metric permits of better managing the overall network performing a better distribution of the calls over each path that goes from source to destination.

This paper is organized as follows: section II presents the related work; in section III the proposed algorithm is described; section IV presents the performance evaluation conducted through an ad-hoc simulator written in C++ language; at last, in section V are summarized the conclusions.

2 Related Works

The number of successful SI applications is growing fast, especially in combinatorial optimization, communicational networks, and robotics.

Two of the most important SI techniques currently in existence are Ant Colony Optimization (ACO) [9], [10], [11] and Particle Swarm Optimization (PSO) [12]. ACO is a meta-heuristic optimization algorithm that can be used to find approximate solutions to difficult combinatorial optimization problems. In ACO artificial ANTs build solutions by moving on the problem graph and they, mimicking real ANTs, deposit artificial pheromone on the graph in such a way that future artificial ANTs can build better solutions. ACO has been applied successfully to an impressive number of optimization problems. PSO is a global minimization technique for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space.

A very successful example of swarm-based routing algorithms is the AntNet adaptive agent-based routing algorithm [6]. AntNet is one of the pioneering approaches for routing with mobile agents. It is inspired by the observation of ANT colony behavior that cooperates to achieve the common goal of finding food. The agents are modeled to emulate the ANTs pheromone technique for finding the shortest path.

In [18] the authors have presented a Multi-Constraints Routing Algorithm based on Swarm Intelligence over High Altitude Platforms and they have shown the convergence of the algorithm. They have carried out many simulations in order to show that the algorithm converges to a sub-optimal solution varying some factors as the number of forward ANTs sent in the network, the decaying factor, the delay bound requested by the user applications.

However, a number of swarm-based routing protocols exist (ARAMA [13], ARA [14], ADRA [15], ARAAI [16], NOPAR [17].

3 ANT Routing Proposed Algorithm

In this paper the proposed algorithm inherits some behavior of AntNet routing but with the extensions of novel metrics for the multi-objective optimization [6], [18]. In particular, we have considered the ANT colonies behavior in order to resolve the problem of finding the optimal route in a telecommunication network that try to minimize not only the path length but also the maximum end-to-end delay bound.

The considered scenario is a HAPs mesh network because these stratospheric platforms are adapted to ANT-based approach. In fact, the HAPs dynamicity permit to set up rapidly a network that can be increased or decreased on the basis of traffic conditions that change quickly in situation of disaster recovery o rescue operations and so on. In order to perform a QoS Routing [19] that guarantees data packets delay lower than maximum end-to-end delay bound we have introduced in the network an Integrated Services (IntServ) architecture with RSVP protocol [20] in order to build in the setup phase an initial solution space. Unlike the classic IntServ, the protocol proposed uses PATH packets to test the network status and RESV packets to find multiple paths that meet minimum quality requirements.

For each RESV arrived at the source node a Forward ANT (FANT) message is generated in order to explore the network to find the better solution.

This algorithm is characterized by the following steps: 1) PATH message is flooded from source HAP to destination HAP. 2) For each PATH message received a destination HAP generates a RESV message that goes through the reverse path of

PATH message. 3) For each RESV arrived to source node, generates a FANT message in order to explore optimum paths. 4) For each FANT arrived the destination node generates a Backward ANT (BANT) message, which follows the path of reverse FANT, updating the probability of selecting the link and the amount of pheromone left on them. The formula for updating the link probability is applied to each link of the HAP crossed by ANT and is shown in the following:

$$p_{i,j} = \frac{\left[\tau_{i,j}\right]^{\alpha} \cdot \left[\eta_{i,j}\right]^{\beta}}{\sum_{j \in N_i} \left[\tau_{i,j}\right]^{\alpha} \cdot \left[\eta_{i,j}\right]^{\beta}} \cdot \quad (1)$$

Where $\tau_{i,j}$ represents the pheromone quantity released by BANT, $\eta_{i,j}$ represents the value of the local heuristic evaluated at the ANTs traveling, α and β represent respectively the pheromone scale factor and visibility factor associated. Both these value are equal to 1. While the ANT can explore multiple paths to find optimum path, according to a probabilistic logic, the data packets follow a deterministic linked to the maximum probability of selecting the links. The data packets are forwarded in the network used the following formula:

$$nextHAP_i^D = \arg\max_{j \in N_i} \left\{ \left[\tau_{i,j}{}^{D} \right] \cdot \left[\eta_{i,j} \right] \right\} . \quad (2)$$

Where N_i represents the set of neighbors nodes to the *i-th* HAP, $nextHAP_i^D$ represents the next HAP with respect to *i-th* HAP in order to reach the HAP Destination (HAP D). This algorithm will use three multiple metrics. In following, these will be described.

3.1 Minimum Hop Count Metric

ANTs that travel in the network leave a pheromone quantity in order to communicate with other ANTs. In particular, the pheromone quantity will be increased in links crossed by FANTs and BANTs and decreased in other links. In the following the pheromone updating formulas are shown.

$$\tau_{i,j}^D(n) = f\left(\rho_{i,j}^D\right)\tau_{i,j}^D(n-1) + g\left(\rho_{i,j}^D\right) . \quad (3)$$

With $j \in N_i$ and with link $(i,j) \in P_{S-D}(n)$.

$$\tau_{i,j}^D(n) = f\left(\rho_{i,j}^D\right)\tau_{i,j}^D(n-1) . \quad (4)$$

With $j \in N_i$ and link $(i,j) \notin P_{S-D}(n)$.

Where $\rho_{i,j}^D$ indicates the *path grade* calculated on the HAP D, $\tau_{i,j}^D(n)$ is the pheromone quantity released by *n-th* ANT for going towards the *j-th* HAP from the *i-th* HAP in order to reach the HAP D, $P_{S-D}(n)$ is the path from the HAP Source (HAP S) to HAP D. *f* and *g* are two functions that depend on the *path grade*. In particular, function *f* represents the "vanishing factor" and *g* represents an "enforcing function":

$$f(\rho) = 1 - \rho \ . \tag{5}$$

$$g(\rho) = \rho^k \ . \tag{6}$$

where $0 \le \rho \le 1 \Rightarrow 0 \le f(\rho) \le 1, 0 \le g(\rho) \le 1$ and $k \in N$ represents the *decaying factor*. The higher *k*, the lower is the contribute of enforcement function, because *the path grade* is $0 \le \rho \le 1$. In the evaporation and enforcing function the *path grade* ρ becomes important. In this work, in accordance with [13], the *path grade* is a function of two indexes $I_{best P_{S-D}}(n)$ and $I^{tot}_{P_{S-D}}(n)$:

$$\rho = F\left(I^{tot}_{P_{S-D}}(n), I_{best P_{S-D}}(n)\right). \tag{7}$$

For more details see [18].

3.2 Maximum Delay Bound Metric

Also for the Maximum Delay Bound metric, an index I_i was introduced associate with each HAP *i-th*. This index can account for the queuing delay on the link *(i,j)* observed by the FANT in the path from HAP S to the HAP D. It can be expressed by the following formula:

$$I_i = \frac{\left(PD_{i,j} + DB_{i,j}\right)}{\sum_{j=1}^{|N_i|}\left(PD_{i,j} + DB_{i,j}\right)} \ . \tag{8}$$

where $PD_{i,j}$ and $DB_{i,j}$ represent respectively the propagation and the queuing delay for the link $(i,j) \in P_{S-D}$. Moreover N_i is the set of HAP near to *i-th* HAP. It is clear that I_i can assume values between 0 and 1. Let I_i^D indicate the normalized index associated with the cost of *i-th* HAP that leads to an HAP D and considering the normalized index $\left[I^{tot}_{P_{S-D}}(n)\right]_D$ associated with the path crossed by FANT and calculated at the receiver, the following expression can be applied to calculate the cost associated with the path discovered by the ANT packets:

$$\left[I^{tot}_{P_{S-D}}(n)\right]_D = \prod_{i=1}^{L} I_i^D = \prod_{l=1}^{L} \frac{\left(PD_{i,j} + DB_{i,j}\right)_l}{\left[\sum_{j=1}^{|N_i|}\left(PD_{i,j} + DB_{i,j}\right)\right]_l} \ . \tag{9}$$

After the calculation of this index, it is possible to determine the *path grade* (for details see [18]). In this case the W ANTs window is always used to determine the best path found in the past by the last W ANTs. In this case, the $I^W_{best P_{S-D}}(n)$ can be computed as follows:

$$I^{W}_{bestP_{S-D}}(n) = \begin{cases} \dfrac{D \cdot \left\{ \max\left[q_{length} \right]_{i,j} + PD_{i,j} \right\}}{N \cdot \left\{ \max\left[q_{length} \right]_{i,j} + PD_{i,j} \right\}} = \dfrac{D}{N} & \text{if W=1} \\ \min\left(I^{W-1}_{P_{S-D}}, \dfrac{D}{N} \right) & \text{if W>1} \end{cases} . \quad (10)$$

With $0 \le \frac{D}{N} \le 1$ where D represents the network diameter and N is the number of nodes considered in the network. The numerator of the expression above represents the maximum cost associable with a path considering that, in the worst situation, a FANT traverses the network diameter. $\max\left[q_{length} \right]_{i,j}$ represents the maximum queuing delay that can be observed on the link (i,j)'s queue of a pre-set length q_{length} .On the other hand, the denominator represents the maximum delay observable if the FANT goes through all the network nodes.

3.3 Traffic Load Distribution Metric

In the case of traffic load distribution another index has been defined to equally distribute the data traffic on the links associated to the HAP belonging to path from a source S to a destination D. This index takes into account the bandwidth availability on the link *(i,j)* that goes from HAP *i-th* to HAP *j-th*. This bandwidth availability can be defined utilizing the concept of link utilization. Let $U_{(i,j)}$ the link utilization of the link *(i,j)* that goes from HAP *i-th* to HAP *j-th*, the bandwidth availability of the link *(i,j)* defined as $bA_{(i,j)}$ can be expressed as following:

$$bA_{(i,j)} = 1 - U_{(i,j)}. \quad (11)$$

In particular, for each HAP *i-th* a value I_i has been considered as follows:

$$I_i = bA_{(i,j)}. \quad (12)$$

This index can assume values in the range *[0,1]* since the $U_{(i,j)}$ is a values between 0 and 1. Now it is possible to define the normalized index $\left[I^{tot}_{P_{S-D}}(n) \right]_D$ associated with the path crossed by FANT and calculate at the receiver. It is expressed as follows:

$$\left[I^{tot}_{P_{S-D}}(n) \right]_D = \min_{i \in P_{S-D}} (I_i) . \quad (13)$$

Now, after the calculation of this index, it is possible to determine the *path grade* (for details see [18]). In this case the W ANTs window is always used to determine the best path found in the past by the last W ANTs. In this case, the $I^{W}_{best\,P_{S-D}}(n)$ can be computed as follows:

$$I^{W}_{best\,P_{S-D}}(n) = \max\left(\left[I^{tot}_{P_{S-D}}(n) \right]_D \right) . \quad (14)$$

Where max operator means that the destination receives more than one index related to different path between source and destination and then it choices the path with the maximum bandwidth availability in order to maximize the network utilization. This index will allow a better traffic distribution because low congested regions (paths) will offer higher bandwidth availability.

Finally, once defined all three metrics we can define a total index that takes into account the minimum hop count metric, the minimum hop count metric with maximum end-to-end delay metric and the load balancing metric. It is possible to define this total index as follows:

$$I_{tot} = C_1 \cdot I_1 + C_2 \cdot I_2 + C_3 \cdot I_3 \ . \tag{15}$$

Where we define I_1 the index of the minimum hop count metric, I_2 the index of the minimum hop count metric with maximum end-to-end delay and I_3 the index of load balancing. Moreover, each of these indexes is multiplied for a constant C_i with $i \in [1,3]$. The constant value C_i can assume values in the range $[0,1]$ and it has to satisfy this condition:

$$\sum_{i=1}^{3} C_i = 1 \ . \tag{16}$$

This constant gives the possibility of taking into account one, two or all three metrics contemporarily, or it gives the possibility of associating a different weight to the metric on the basis of the value that can be assigned to constant C_i.

4 Performance Evaluation

Many simulations were carried out in order to demonstrate the goodness of this proposed routing algorithm. In fact, it is possible to note that it is capable of finding a suboptimal solution. In table 1 the algorithm parameters common to all the simulation campaigns are summarized.

Table 1. Algorithm Parameters

Algorithm Parameters	Value
W (window size)	10
A	20
τ	0.4
k (decaying factor)	3
ANT Rate	500(ms)

4.1 Simulation Scenario

In this work we have considered a network composed of 13 HAPs interconnected between them in order to create a HAPs mesh. where the link labeled with the IHL - InterHapLink identification numbers are the links that have been under observation

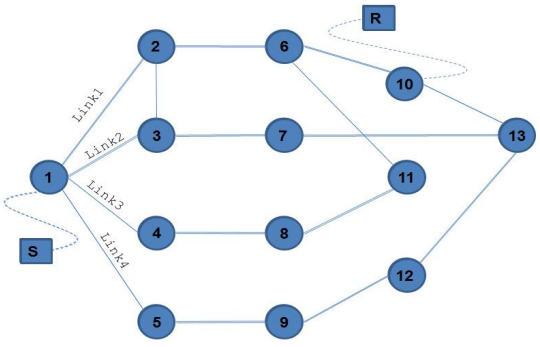


Fig. 2. Simulation Topology

(see Fig. 2.). Moreover, a RTT of 0.4 ms, forward and return channel of 469 slots, trama of 47 ms, atomic channel of 32 Kbps and maximum number of 16 sources for type D RCST was considered. We have presented a comparison of more than one algorithm in order to show the goodness of the ANT routing algorithm in terms of signaling overhead and traffic distribution. In the following graphs of some performed simulations are shown.

4.2 Simulation Results

We have performed some simulations in order to compare a simple Shortest Path Algorithm (SPA) with three variants of our algorithm: ANT Routing with Minimum Hop Count (A_MHC), ANT Routing with Minimum Hop Count and Maximum end-to-end Delay (A_MHC_MD) and ANT Routing with Minimum Hop Count, Minimum Hop Count with Maximum end-to-end Delay and Load Balancing (A_LB) metrics. In future works we will show a comparison between our algorithms with other QoS routing algorithms.

We have considered these variants in order to show how the simple ANT Routing with the Minimum Hop Metric have a behavior similar to a classical SPA and how an algorithm that considers the three metrics contemporarily has the best performance because it is capable of distributing the traffic over all paths that go from the source to the destination without congesting any network regions. In this simulation campaign we have considered variation of traffic load between light and high traffic load condition. In particular, we have considered as light traffic load a rate of 0.4 calls/minute instead we have considered as high traffic load a rate of 2 calls/minute. In the first case the links in the network reach 10%-15% of their utilization whereas in the second case the links utilization can reach a percentage of 100%.

Fig. 3. and Fig. 4. show the number of accepted calls in light and heavy traffic load conditions. The graphics show a comparison between four algorithms. It is possible to note observing Fig. 3. that with a requested Delay Bound (DB) of 600 ms the SPA algorithm and the A_MHC present similar performance. Specifically, they accept the same number of calls in the system. Both algorithms find the minimum path between source and destination on which are routed all packets until path saturation. Considering the A_MHC_MD algorithm it is possible to note that the system accepts a greater

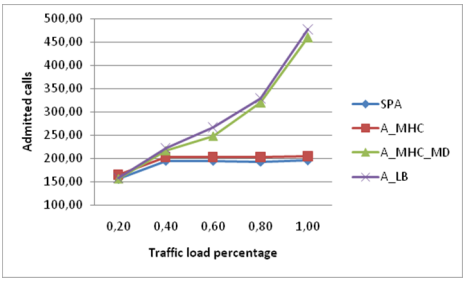


Fig. 3. Number of admitted calls vs. traffic load percentage with DB=600

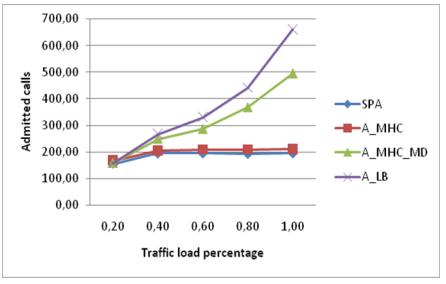


Fig. 4. Number of admitted calls vs. traffic load percentage with DB=[500,1000]

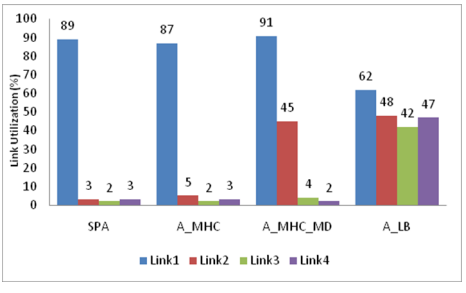


Fig. 5. Link utilization vs. four routing algorithms

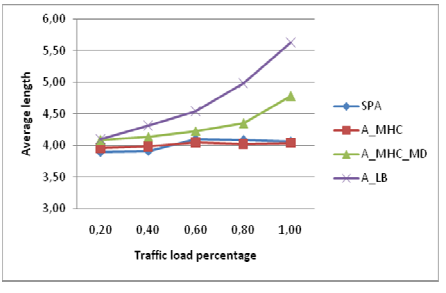


Fig. 6. Average length vs. traffic load percentage

number of calls with the increasing of traffic load condition. This graphic shows that the performance of the A_LB algorithm are similar to the previous algorithms. Both algorithms manage more calls than the other two simple algorithms because the added metrics improve the resource system management. Fig.4. shows the four algorithms comparison considering a requested DB varying between 500 and 1000 ms. It is possible to note that yet the SPA and A_MHC have a similar performance because they saturate the shortest path and then they refuse each incoming call. The A_MHC_MD algorithm, as we said before, admits a greater number of calls because after having saturated the minimum path it considers the second shortest path that guarantees the required DB. In this graphic it is also possible to view how the A_LB algorithm has a better admitted calls number trend. This is due to the consideration of the third metric that guarantees a call distribution on all valid paths between source to destination. In this case, in fact, we have considered a range of DB required by receivers. This permits to better manage calls between all possible paths. We recall that great DB permits paths with more hops respect to small DB that instead permits paths with lower number of hops.

In order to show the utilization links grade using four algorithms we have considered only a source-destination couple that requires a DB of 500 ms. In particular, we have considered only the first link of each path of the source-destination couple. A couple with this DB can find 6 paths that satisfy the DB constraint. In particular, the total paths that can be identified in the graph of the simulation topology are 7:

Path1={1,2,6,10}, Path2={1,3,7,13,10}, Path3={1,3,2,6,10}, Path4={1,2,3,7,10,13}, Path5={1,4,8,11,6,10}, Path6={1,5,9,12,13,10}, Path7={1,5,9,12,13,7,3,2,6,10}.

As it is possible to note there are one path of 4 hops that represents the minimum path length, two paths of 5 hops, three paths of 6 hops and finally one path of 10 hops. All identified paths satisfy the DB requirement except that of 10 hops that is discarded because it does not respect the maximum delay metric. As it is possible to view the first links of the paths can be shared.

Fig. 5. shows how the SPA algorithm utilizes only the minimum path. The A_MHC has a similar behavior. Instead the other two algorithms present different behaviors. Specifically, A_MHC_MD that uses two metrics, after having saturated the minimum path, considers the second minimum path that guarantees the DB requirement. As it is possible to view in the figure. Differently is the behavior of the fourth algorithm, it uses all the paths that satisfy the QoS requirements permitting a traffic load distribution. The A_LB assures a better resource management avoiding to saturate links on minimum hop count path that can be useful to manage call with reduced delay bound. Moreover, the utilization metric of A_LB improves the network utilization increasing the number of accepted calls such as previously explained in Fig. 4.

Finally, we show another graphic, Fig.6., in which the average path length is depicted. It is possible to note how with light traffic load all four algorithms have similar behavior. They have an average path length of about 4 hops that are the hops of the minimum path. When the system is in a heavy traffic load condition the average path length presents a different trend.

The SPA and A_MHC have the same previous trend, instead the algorithms that take into account also the QoS metrics of delay and traffic load balancing have an average path length greater. This is due to the different algorithms behavior. In the case of the A_MHC_MD algorithm when the minimum path is saturated it considers other paths that satisfy the QoS metrics and then permit to better utilize the system resource, in particular it considers the second shortest path. Instead, the A_LB algorithm permits a better resource management because it allows to select different paths on the basis of the bandwidth availability. Thus also paths with a greater number of hops and with higher utilization can be selected increasing the average path length such as depicted in Fig. 6.

5 Conclusions

In this paper we have proposed a novel algorithm based on Swarm Intelligence. We have chosen as reference architecture a HAPs mesh network in order to take advantage of some useful characteristics such as high bandwidth, lower propagation delay and dynamism. This algorithm permits to discover minimal length paths that satisfy requirements of end-to-end delay and/or bandwidth taking into account also the traffic load distribution. The proposed algorithm has been realized using a novel technique called ANT Routing that is based on the ANT colony behavior. In particular, this technique results particularly important in the resolution of optimization problems. The proposed algorithm works to satisfy multiple metrics associated with path length, maximum end-to-end delay and load balancing. Innovation factor for the proposal has been the pheromone updating policy. Many simulation campaigns have been

performed in order to evaluate the goodness of our proposed algorithm. We have performed a comparison between our algorithm with one, two and three metrics and a Shortest Path Algorithm in order to show how the ANT based algorithm with all three metrics is capable of better utilizing the system resource distributing the traffic load over the network without creating congested regions.

References

1. Tozer, T.C., Grace, D.: High Altitude Platforms for Wireless Communications. Electronics & Communication Engineering Journal (June 2001)
2. De Rango, F., Tropea, M., Marano, S.: Integrated Services on High Altitude Platform: Receiver Driven Smart Selection of HAP-Geo Satellite Wireless Access Segment and Performance Evaluation. International Journal on Wireless Information Networks 13(1), 77–94 (2006)
3. Nedjah, N., de Macedo Mourelle, L.: Swarm Intelligent Systems. Springer, Heidelberg (2006)
4. Garnier, S., Gautrais, J., Theraulaz, G.: The biological principles of swarm intelligence. Swarm Intelligence Journal (2007)
5. Dempsey, P., Schuster, A.: Swarm Intelligence for network routing optimization. Journal of Telecommunications and Information Technology (2005)
6. Di Caro, G., Dorigo, M.: AntNet: Distributed Stigmergetic Control for Communications Networks. Journal of Artificial Intelligence Research (JAIR) 9, 317–365
7. Sun, J., Xiong, S., Guo, F.: A new pheromone update strategy in Ant Colony Optimization. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics (August 2004)
8. Lorenz, D.H., Raz, D.: A simple efficient approximation scheme for the restricted shortest path problem. Oper. Res. Lett. 28(5), 213–219 (2001)
9. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
10. Sim, K.M., Sun, W.H.: Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions. IEEE Transactions on System, Man and Cybernetics Part A: Systems and Humans 33(5) (September 2003)
11. Dorigo, M., Birattari, M., Stutzle, T.: Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique. IEEE Computational Intelligence Magazine 1(4), 28–39 (2006)
12. Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications and resources [A]. In: Proceedings of the 2001 Congress on Evolutionary Computation [C], Seoul, South Korea, pp. 81–86 (2001)
13. Hussein, O.H., Saadawi, T.N., Jong Lee, M.: Probability Routing Algorithm for Mobile Ad Hoc Networks' Resources Management. IEEE Journal on Selected Areas in Communications (2005)
14. Gunes, M., Sorges, U., Bouazizi, I.: ARA - The Ant-Colony Based Routing Algorithm for MANETs. In: Proceedings of International Conference on Parallel Processing Workshops. IEEE, Los Alamitos (2002)
15. Zheng, X., Guo, W., Liu, R.: An Ant-Based Distributed Routing Algorithm for Ad-hoc Networks. In: International Conference on Communications, Circuits and Systems (ICCCAS). IEEE, Los Alamitos (2004)
16. Y.-Yuan, Z., Yan-xiang, H.: Ant Routing Algorithm for Mobile Ad-hoc Networks Based on Adaptive Improvement. School of Computer Science, Wuhan University School of Computer and State Key Lab of Software Engineering, Wuhan University Wuhan, Hubei Province, China (2005)

17. Baguenine, F., Mellouk, A.: N-Best Optimal Path Ant Routing Algorithm for State-Dependent N Best Quality of service Routes in IP Networks. In: 32nd IEEE Conference on Local Computer Networks (LCN 2007), pp. 747–754 (2007)
18. De Rango, F., Tropea, M., Provato, A., Santamaria, A., Marano, S.: Multi-Constraints Routing Algorithm based on Swarm Intelligence over High Altitude Platforms. In: NICSO 2007 International Workshop on Nature Inspired Cooperative Strategies for Optimization, Acireale, Italy, November 8-10 (2007)
19. Bashandy, A.R., Chong, E.K.P., Ghafoor, A.: Generalized Quality-of-Service Routing With Resource Allocation. IEEE Journal on Selected Area of Comm. 23(2) (February 2005)
20. White, P.: RSVP and integrated services in the Internet. A tutorial, IEEE Communications Magazine (May 1997)
21. Arabshahi, P., Gray, A., Kassabalidis, I., Das, A.: Adaptive Routing in Wireless Communication Networks using Swarm Intelligence. In: 9th AIAA Int. Communications Satellite Systems Conf., Toulouse, France, April 17-20 (2001)
22. Tanenbaum, A.S.: Reti di Computer. UTET

ECA-Rule Visual Programming for Ubiquitous and Nomadic Computing

José Oscar Olmedo-Aguirre[1], Mónica Rivera de la Rosa[2], and Guillermo Morales-Luna[3,⋆]

[1] Electrical Engineering, CINVESTAV-IPN, Mexico City, Mexico
ooolmedo@cs.cinvestav.mx
[2] Escuela Superior de Cómputo, ESCOM-IPN, Mexico City, Mexico
[3] Computer Science, CINVESTAV-IPN, Mexico City, Mexico
gmorales@cs.cinvestav.mx

Abstract. System modeling, analysis and visualization are becoming a common practice for the design of distributed intelligent systems since the wide adoption of the Unified Modeling Language (UML). However, UML cannot describe important behavioral properties such as context awareness as required for ubiquitous computing. In this paper, we present *Context Aware UML Sequence diagrams* (CA UML-S), an experimental visual programming language that extends UML sequence diagrams with data/object spaces to represent computational context awareness. The programming language provides the means to describe the event-condition-action (ECA) rules that govern complex nomadic user behavior and to visualize their effect. The ECA rules are compiled into common concurrent programming abstractions by introducing structuring notions of object creation, synchronization, and communication, along with sequential and selective composition of simpler rules. The contribution of this work is in providing programming abstractions that facilitate the design of context-aware applications for ubiquitous and nomadic computing.

1 Introduction

Mark Weiser introduced the term *ubiquitous computing* [1] to describe an environment where a fixed computational and communication infrastructure will help nomadic users to conduct their activities unconscious of the infrastructure. However, the design and construction of context-aware applications in the highly dynamic setting of ubiquitous computing has been recognized as a difficult undertaking. In this respect, rule-based programming is proposed to improve our understanding of complex program behavior. Rule-based specifications are used in reactive systems that are monitored for the occurrence of events that may signal critical conditions. In rule-based systems, once an event is detected, if the event parameters satisfy certain condition, a specified action is performed to handle the situation. The language construct that corresponds to this model of

⋆ Olmedo and Morales acknowledge the partial support of Mexican Conacyt.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 925–935, 2008.

interaction is called *ECA-rule* (*Event-Condition-Action–rule*). Such rules arise from the fact that a distributed system is a system whose behavior is driven by the events of receiving messages and by the actions of sending messages. ECA-rules can be adopted as the basis for data representation, message exchange, and coordination specification in distributed applications [2]. Rule-based systems may as well evolve into complete programming languages aimed to coordinate software agents with proactive, rational and social behavior. ECA-rules constitute a powerful programming paradigm, though their interactions are very often difficult to understand because they are generally not designed to be applied in a structured or hierarchical manner.

With the purpose to make rule interactions easier to understand, we propose to use UML sequence diagrams to visualize rule definition, selection and execution. We believe that rule definition can be easily understood by means of a short UML sequence diagram that exposes the event-condition-action structure of the rule. Furthermore, rule selection and instantiation becomes more coherent as the collection of sequence diagrams are identified by the events they handle. Rule execution may become more understandable because rules are grouped around objects in a sequential or parallel manner.

Unfortunately, the programming environments that have adopted UML diagramming as the basis for its design methodology are used at best for the generation of program skeletons in programming languages as `Java` and `C++` just because UML diagrams cannot be directly executed. Although replacing the textual representation of a programming model for a visual one represents a substantial step in abstracting algorithm design from program coding, program behavior still cannot be visualized directly from UML diagrams as the program execution is not integrated into the environment.

We propose the construction of a visual programming language that addresses the afore-mentioned problems. By integrating visualization in the program specification and execution, the programming language offers a number of outstanding advantages from previous work:

- Priming algorithmic design over program coding
- Abstracting algorithms from programming languages and platforms
- Specifying and visualizing program behavior with UML sequence diagrams

We outline the content of this paper. In section 2, we compare our proposal with some related work in the areas of program visualization and formal methods used. In section 3, fundamental concepts of Context Aware UML-S are presented and illustrated with an example of user guiding from nomadic computing. The visual specification and execution of ECA rules is discussed and analyzed in section 4. The programming model in the structured operational semantics style is presented in section 5. Finally, in section 6, we present our concluding remarks.

2 Related Work

Visualization of concurrent programs is more complicated than visualization of sequential programs, due to the presence of multiple communicating threads,

establishing a competition for resources, and requiring periodical synchronization. The misunderstanding of concurrent programs behavior may result in unexpected interactions and non-deterministic executions. However, some visualization tools have been realized to overcome these issues. In the `Gthreads` library [3] a program graph is built as threads are forked and functions are called. The vertices of the graph represent program entities and events, while the edges represent temporal orderings among vertices. Message passing views are supported by the `Conch` system [4] where processes appear outside of a ring and messages are exchanged among them by traversing the ring. In this way, undelivered messages can be detected as they remain within the ring. The `Hence` system [3] offers animated views of the program graph obtained from execution of PVM programs. None of them use standard UML diagrams to describe process interactions. Programming and visualization of sequential programs using an UML-like visual programming language has been explored in `Moon` [5], which is the base of the current prototype implementation of CA UML-S diagrams.

Recently formal models based on Petri nets have been applied to ubiquitous computing due to their intuitive graphical representation and well established theoretical foundation. An *Elementary Object System* (EOS) [6] is a two-level net description consisting of a *system net* marked by high-level structured tokens representing *object nets*. Object nets can change their marking but not their structure, following the principles of object-oriented modeling. EOS has been used to model mobile user behavior. Further refinement of EOS are *Nested Petri Nets* (NPN) [7] which are multi-level net descriptions to model hierarchical multi-agent dynamics. In NPN, object nets have their own structure and behavior, being able to appear, to evolve or to disappear during lifetime, and their number is unlimited. An NPN has four steps: `transfer`, `object-autonomy`, `horizontal synchronization` and `vertical synchronization` steps. A `transfer` step can move, generate or remove objects with no change of their inner states. An `object-autonomy` step can change the inner state of one object. `horizontal synchronization` conveys simultaneous firing of two object nets situated in the same place of a system net. `vertical synchronization` means simultaneous firing of the system net with one of the object nets involved in the firing. *Ubiquitous Nets* (UN) [8] allows modeling both devices (processor nets) and software components located at processors (process nets). In the model, processes change their location due to the firing of special movement transitions.

3 CA UML-Sequence Visual Programming Language

UML-sequence (UML-S) diagrams describe the behavior of a system developed by the interaction of the participants. These diagrams contain objects (*classes*) that exchange *messages* arranged in a *time sequence*. They are defined according to *class roles*, *lifelines, activations* and *messages* [9]. Illustrative examples of these diagrams are shown in [10]. *Class roles*, denoted by rectangles surrounding the role-name and the class-name, specify the type of objects that may participate with interactions and collaborations. *Lifelines*, denoted by vertical dashed lines,

represent the existence of class roles over a period of time, since the object creation until its destruction. They may split into concurrent lifelines to show concurrency or conditionality, each corresponding to a thread or a conditional branch and they may merge together at some subsequent point. *Activations*, denoted by thin vertical rectangles arranged along lifelines, represent the time during which a class role is performing an action (*operation*) or when it is *active* and has *focus of control*. Finally, *messages*, denoted by labeled horizontal arrows between lifelines, define the information content of a communication that is exchanged in interactions and collaborations. The message instance has a *sender*, a *receiver* and possibly other information according to the characteristics of the request. Messages may be either *synchronous* (solid arrow heads), with explicit or implied *return* (stick arrow heads), or *asynchronous* (half stick arrow heads).

Upon standard UML-S diagrams, new diagrammatic elements are introduced, among explicit program states, data/object spaces, computational context and context awareness, in order to define *Context Aware* UML-S or CA UML-S visual programming language. *Explicit program states* (*configurations*) are represented by graphic elements similar to those used for *class roles* except that they appear along the *timeline* possibly with a note containing an assertion about the program state. Explicit program states are useful because they can simplify the rendering of visual programming constructions to reflect the change of the object configuration. *Data spaces* contain heterogeneous collections of pieces of information that are stored in a single or a networked cluster of computational devices. *Object spaces* contain data communication enabled objects that can be detected if they are within the scope of a short range local area network, and can be identified by the physical coordinates of the communication base of the network. *Data/object spaces* can be seen, in abstract terms, as partitions of the entire data collection and the objects found in the environment. *Data/object spaces* are identified by unique global identifiers. In CA UML-S, a space is represented by a rounded box labeled by the space name. *Computational context* is the entire computation, communication and coordination infrastructure conformed by sensors, actuators, computational, and communication devices that the environment offers to enable users and agents to interact with them and with the environment. Thus, the computational context enables ordinary objects to have a presence in this form of extended reality. *Context awareness* of a mobile user or agent is the ability to perceive this extended reality by identifying the spaces and the objects that populate them. The entities located in a space are identifiable and shared resources that can be accessed by users and agents that are aware of the existence of such a space.

Users and agents are aware of spaces if they know the name that identified them. As proposed in this work, CA UML-S for ubiquitous computing extends the class role notation to accommodate the name of the space to which a class instance interacts. Such a space name can be avoided in the diagram by drawing an arrow pointing to the current space to which the object is aware. Note that space variables describe the local topology of the object neighborhood and can change their values to reflect the local space context awareness.

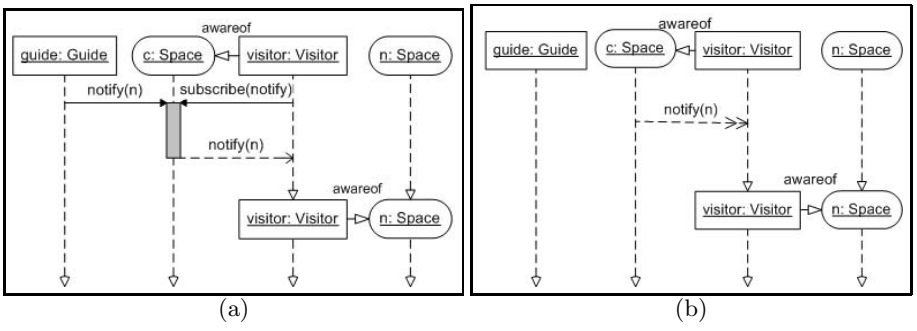


Fig. 1. Guide telling to the locations along a path where the visitor should move on

CA UMLS diagram composition is exemplified with a recurrent mobility problem in nomadic computing about how to guide a visitor to his/her destination. Figure 1 shows a CA UML-S diagram displaying the interaction that the guide maintains with the local communication infrastructure. Assuming the entire region is partitioned into a grid of addressable physical locations, each of such locations comprises a data/object space that contains information about the local topology of its neighbor locations (i.e. north, east, south, etc.). The interaction begins when, either the guide sends to the current location a message telling which neighbor location to move on next or the visitor sends a subscription to the current location to receive the messages from the guide. In particular, in Figure 1(a) the guide sends the message to the visitor local space suggesting to move on to the north and this message is forwarded later on to the visitor. The CA UML-S synchronization (dark gray) bar specifies, with no time-ordering restrictions, all the messages that are expected to be received (incoming arrows) before some responsive actions may take place (outgoing arrows).

The recurrent pattern shown in Figure 1(a) is concisely represented by a double headed arrow as illustrated in Figure 1(b). It is properly defined because forwarding a message from the space to the subscriber can only happen when both the message and the subscription were received before at the space. This simplification can be used only when the message sender is not needed to be known in the rule description. This is a reasonable assumption because data/object spaces were conceived to decouple the interaction among the participants. Besides, the notation emphasizes the more relevant aspects of specifying what data/object is required and from which location should be taken.

Using the Dijkstra's algorithm to find a minimal path between two connected points in a map, the locations lying in the minimal path can be determined. Then the guide places a notification addressed to the visitor in all the locations along the path. The notification indicates in what direction the visitor must move on next to get closer to its destination. The visitor is aware of the data/object space of the current location and places a subscription on it to receive the notification of where to move on. When both, the subscription and the notification become

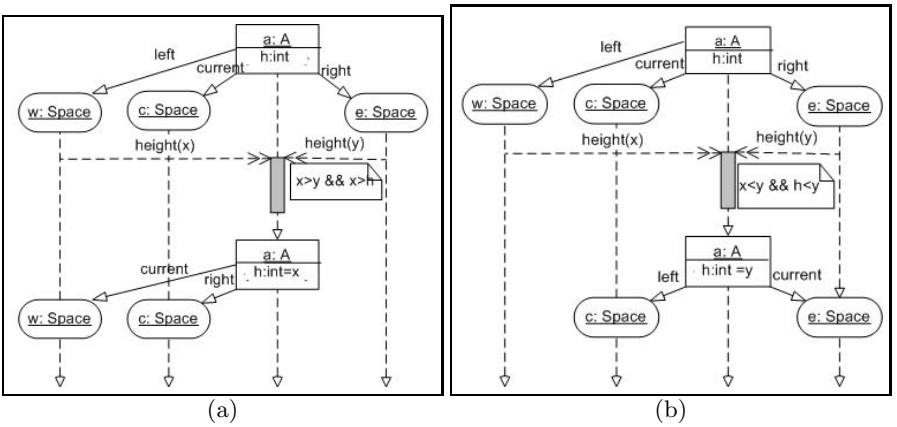
available, the local space forwards the notification to the visitor, who then determines the physical address of the next location. By repeating this process, the locations lying in the path take the visitor to its destination.

Unfortunately UML-S diagrams were conceived having no applications to nomadic computing in mind. Therefore the fundamental concept of locality is not represented in the diagrams, and in consequence this interaction cannot be modeled in UML. Though a partial solution can be devised by using UML stereotypes to define a new entity, this approach neglects the fundamental knowledge representation problem that poses human or process mobility. In our approach, UML-S for ubiquitous computing extends the class role notation to accommodate the name of the location to which a class instance interacts. Such a location name can be avoided by using instead an arrow that points to the current location. Although simple, this solution proves to be expressive enough to model complex interactions in nomadic computing that cannot be represented otherwise in UML.

4 ECA Rule Programming in CA UML-S

The CA UML-S synchronization bar constitutes the fundamental visual construction to define ECA rules in the programming language, because they comprise events, conditions and actions. *Events*, denoted by incoming message arrows with no time-ordering restrictions, are notifications of the execution of observable actions. *Conditions*, denoted by logical expressions written in notes, are constraints defined upon the contents of the events and the object configurations to select those that satify the contranints. *Actions*, denoted by either outgoing message arrows or explicit object configurations, are operations that modify the object state and its awareness of the local context. Figure 2 shows two rules depicted in the CA UML-S visual language using the synchronization bar. Each rule renders two possible scenarios that arise from the conditions that may occur. Even with the few elements presented here, it can be shown that very complex and interesting behaviors can be visually described. Among the experiments we have conducted so far, hypermedia architectures, simple workflow systems, collaborative systems and Turing machines are among the most notorious. Hill Climbing and other search space techniques used in Artificial Intelligence are currently being investigated.

The Hill Climbing problem consists of finding a location containing data or object attributes that maximizes a function. The solution to this problem consists on evaluating the function on the neighbor locations and selecting the location that increases the value of the function. This evaluation, selection and movement process is repeated until a (local) maximum is reached. Figure 2 shows two visual rules that are part of the solution of a bi-dimensional version of the Hill Climbing problem. As the figure shows, the visitor is aware of the current location in which he/she stands up, and the one on the left and the one on the right of the current location. This is depicted by the outgoing arrows from the visitor instance box to the corresponding space instance boxes in the upper



(a) (b)

Fig. 2. A bi-dimensional Hill Climbing program description

part of the drawing. The visitor moves when its current location is changed by any of its two neighbor locations. The selection is taken for the location that contains the greater value obtained from sensor reading of the location height and from object property `h` that keeps the currently known local maximum. The synchronization bar on the visitor object receives message `height` from both locations to be able to make the selection. The restriction note at the right of the synchronization bar describes the selection criteria for each case. As a result, the actions consist in updating the values of the object properties that reflect a change in its context awareness. Fig. 2(a) and 2(b), respectively, shows that the user moves to the left or to the right of the current location.

5 Operational Semantics of CA UML Sequence Diagrams

Formal interpretation of CA UML-S diagrams can be established according to the classic *structural operational semantics* due to Hennessy and Plotkin [11] by defining object configurations and a transition relation over them. An *object configuration* is a triplet $a : (A, \sigma, \varsigma)$ comprising responsive action A, program state σ, and context awareness set ς of data/object spaces. The transition relation states that whenever the object a behaves according to action A at state σ and awareness set ς, then the execution leads to either an object configuration with action A' to perform at state σ' and awareness set ς' or a terminating configuration with state σ' and awareness set ς'.

$$a : (A, \sigma, \varsigma) \rhd a' : (A', \sigma', \varsigma') \quad , \quad a : (A, \sigma, \varsigma) \rhd a' : (\sigma', \varsigma').$$

The transition relation is naturally extended to collections of objects by structural congruence. An object in the collection develops its behavior concurrently and independently from any other, beginning by executing its constructor:

$$\frac{a : (A\,,\,\sigma\,,\,\varsigma) \rhd a' : (A'\,,\,\sigma'\,,\,\varsigma')}{a : (A\,,\,\sigma\,,\,\varsigma)\,,\,O \rhd a' : (A'\,,\,\sigma'\,,\,\varsigma')\,,\,O} \quad \frac{O \rhd O'}{a : (A\,,\,\sigma\,,\,\varsigma)\,,\,O \rhd a : (A\,,\,\sigma\,,\,\varsigma)\,,\,O'}$$

Object behavior evolution is captured in the extended transition relation by including its reflexive and transitive closure. In the description of the SOS semantics, we always refer to the extension of the transition relation which can be seen as a rewriting relation defined upon multisets of object configurations. Under this interpretation, the congruence relation rewrites either a single object configuration or the remaining collection of object configurations.

The abstract syntax of deterministic programs is generated from the following grammar rules of the `Java` subset considered here, consisting of statements among assignments on simple variables, selective instructions, sequential composition, and object method invocations:

$$A ::= \texttt{stop;} \mid \texttt{skip;} \mid x = E; \mid \texttt{if(}B\texttt{)}A_1 \texttt{ else } A_2 \mid A_1\ A_2 \mid o.m(E_1, \ldots, E_n);$$

The class of arithmetic expressions E consists of integer constants, integer variables, and compound expressions formed by composition of arithmetic expressions with usual arithmetic operators. The class of Boolean expressions B consists of Boolean constants, simple Boolean expressions formed by comparison of arithmetic expressions, and composition of Boolean expressions with the logical connectives of negation, conjunction, and disjunction:

$$\begin{aligned} E &::= c \mid v \mid E + E \mid \cdots \\ B &::= \texttt{true} \mid \texttt{false} \mid E\texttt{==}E \mid E \leq E \mid \texttt{!}B \mid B\ \texttt{\&\&}\ B \mid B\ \texttt{||}\ B \end{aligned}$$

When an arithmetic or Boolean expression is evaluated, the result can be defined by extending the introduced transition relation. The evaluation terminates when an arithmetic or Boolean constant is reached, with no changes in the state:

$$a : (E, \sigma, \varsigma) \rhd a' : (E', \sigma, \varsigma) \qquad , \qquad a : (E, \sigma, \varsigma) \rhd E.$$

The value is defined inductively on the syntactic structure of the expression:

$$\frac{}{a : (c\,,\,\sigma\,,\,\varsigma) \rhd c} \qquad \frac{\sigma(x) = c}{a : (x\,,\,\sigma\,,\,\varsigma) \rhd c}$$

$$\frac{a : (E_1\,,\,\sigma\,,\,\varsigma) \rhd c_1 \quad a : (E_2\,,\,\sigma\,,\,\varsigma) \rhd c_2}{\begin{array}{c} a : (E_1 + E_2\,,\,\sigma\,,\,\varsigma) \rhd c_1 + c_2 \\ a : (E_1 \leq E_2\,,\,\sigma\,,\,\varsigma) \rhd c_1 \leq c_2 \end{array}}$$

$$\frac{}{a : (\texttt{true}\,,\,\sigma\,,\,\varsigma) \rhd \texttt{true}} \qquad \frac{}{a : (\texttt{false}\,,\,\sigma\,,\,\varsigma) \rhd \texttt{false}}$$

$$\frac{\begin{array}{c} a : (B_1\,,\,\sigma\,,\,\varsigma) \rhd \texttt{true} \\ a : (B_2\,,\,\sigma\,,\,\varsigma) \rhd a : (b_2\,,\,\sigma\,,\,\varsigma) \end{array}}{a : (B_1\ \texttt{\&\&}\ B_2\,,\,\sigma\,,\,\varsigma) \rhd a : (b_2\,,\,\sigma\,,\,\varsigma)} \qquad \frac{a : (B_1\,,\,\sigma\,,\,\varsigma) \rhd \texttt{false}}{a : (B_1\ \texttt{\&\&}\ B_2\,,\,\sigma\,,\,\varsigma) \rhd \texttt{false}}$$

The structured operational semantics of CA UML-S diagrams given in Table 1 is discussed next. Objects of declared class B are created and initialized by operator `new`, binding the object instance of the class to a variable of the same type and starting its behavior according to the actions given in the class constructor $B()$. Null command `skip;` does nothing, maintaining program state unchanged. Termination command `stop;` ceases object activity. An assignment

Table 1. Structured operational semantics of UML-S diagrams

$$a : (\texttt{Space}\ s = \texttt{newSpace}()\,;\ A\ ,\ \sigma\ ,\ \varsigma) \rhd a : (A\ ,\ \sigma\ ,\ \varsigma\,\{s \mapsto \texttt{create(Space)}\})$$

$$a : (B\ b = \texttt{new}B()\,;\ A\ ,\ \sigma\ ,\ \varsigma) \rhd a : (A\ ,\ \sigma\,\{b \mapsto \texttt{create}(B)\}\ ,\ \varsigma) \quad b : (B(), \emptyset, \emptyset)$$

$$\frac{a : (E\ ,\ \sigma\ ,\ \varsigma) \rhd c}{a : (x = E\,;A\ ,\ \sigma\ ,\ \varsigma) \rhd a : (A\ ,\ \sigma\,\{x \mapsto c\}\ ,\ \varsigma)}$$

$$a : (\textbf{skip}; A\ ,\ \sigma\ ,\ \varsigma) \rhd a : (A\ ,\ \sigma\ ,\ \varsigma)$$

$$a : (\textbf{stop};\ ,\ \sigma\ ,\ \varsigma) \rhd a : (\sigma\ ,\ \varsigma)$$

$$\frac{a : (A_1\ ,\ \sigma\ ,\ \varsigma) \rhd a : (A_2\ ,\ \sigma'\ ,\ \varsigma')}{a : (A_1\ A\ ,\ \sigma\ ,\ \varsigma) \rhd a : (A_2; A\ ,\ \sigma'\ ,\ \varsigma)}$$

$$\frac{a : (A_1\ ,\ \sigma\ ,\ \varsigma) \rhd a : (\sigma'\ ,\ \varsigma')}{a : (A_1\ A\ ,\ \sigma\ ,\ \varsigma) \rhd a : (A\ ,\ \sigma'\ ,\ \varsigma')}$$

$$\frac{a : (B\ ,\ \sigma\ ,\ \varsigma) \rhd a : \texttt{true}}{a : (\texttt{if}\ \ (B)\ \ A_1\ \texttt{else}\ \ A_2\ ,\ \sigma\ ,\ \varsigma) \rhd a : (A_1\ ,\ \sigma\ ,\ \varsigma)}$$

$$\frac{a : (B\ ,\ \sigma\ ,\ \varsigma) \rhd a : \texttt{false}}{a : (\texttt{if}\ \ (B)\ \ A_1\ \texttt{else}\ \ A_2\ ,\ \sigma\ ,\ \varsigma) \rhd a : (A_2\ ,\ \sigma\ ,\ \varsigma)}$$

$$\frac{S' = S\,.\,\texttt{subscription.put}(\texttt{typeof}(x), a)}{a : (s\,.\,\texttt{subscribe}(\texttt{typeof}(x)); A\ ,\ \sigma\ ,\ \varsigma\,\{s \mapsto S\}) \rhd a : (A\ ,\ \sigma\ ,\ \varsigma\,\{s \mapsto S'\})}$$

$$\frac{S' = S\,.\,\texttt{collection.put}(\texttt{typeof}(E), E\sigma)}{a : (s\,.\,\texttt{send}(E); A\ ,\ \sigma\ ,\ \varsigma\,\{s \mapsto S\}) \rhd a : (A\ ,\ \sigma\ ,\ \varsigma\,\{s \mapsto S'\})}$$

$$\frac{S' = S\,.\,\texttt{collection.put}(\texttt{typeof}(M(\overline{E}, \overline{S})), M(\overline{E}, \overline{S})\,\sigma\,\varsigma)}{a : (s\,.\,\texttt{send}(M(\overline{E}, \overline{S})); A\ ,\ \sigma\ ,\ \varsigma\,\{s \mapsto S\}) \rhd a : (A\ ,\ \sigma\ ,\ \varsigma\,\{s \mapsto S'\})}$$

$$\frac{\begin{array}{c} T = \texttt{typeof}(x) \\ a \in S\,.\,\texttt{subscription.get}(T) \qquad (E, S') = S\,.\,\texttt{collection.get}(T) \end{array}}{a : (s\,.\,\texttt{receive}(x); A\ ,\ \sigma\ ,\ \varsigma\,\{s \mapsto S\}) \rhd a : (A\ ,\ \sigma\,\{x \mapsto E\}\ ,\ \varsigma\,\{s \mapsto S'\})}$$

$$\frac{\begin{array}{c} T = \texttt{typeof}(M(\overline{x}, \overline{z})) \\ a \in S\,.\,\texttt{subscription.get}(T) \qquad (M(\overline{E}, \overline{S})), S') = S\,.\,\texttt{collection.get}(T) \end{array}}{a : (s\,.\,\texttt{receive}(M(\overline{x}, \overline{z})); A, \sigma, \varsigma\,\{s \mapsto S\}) \rhd a : \left(A, \sigma\,\left\{\overline{x \mapsto E}\right\}, \varsigma\,\left\{s \mapsto S', \overline{z \mapsto S}\right\}\right)}$$

$x = E$; into a simple variable evaluates expression E at the right-hand side and binds this value to variable x at the left-hand side, updating program state σ to $\sigma\{x \mapsto E\sigma\}$. Sequence of commands $A_1; A$ starts executing A_1 at state σ and upon its completion continues executing command A. Conditional execution `if` `(` B `)` A_1 `else` A_2 selects the execution of either action A_1 or action A_2 , as condition B is evaluated `true` or `false`, respectively.

A data/object space, created by operator `new` upon generic container class `Space`, behaves according to the publish/subscribe pattern. In a data space, a subscription occurs when an object request is received to get any data available by calling method `subscribe`, whereas a publication occurs when an object posts data in the space by calling method `send`. However, regardless of the order of their occurrence, having received in the space a subscription and a publication for the same data type, the space forwards the message published to the object registered in the subscription. Then the subscriber receives such a message by calling method `receive`. Like data spaces, object spaces interact through the same basic methods `subscribe`, `send` and `receive`, leading to similar behaviors. However, object space methods are more elaborated in the message structure and content to pass not only simple data but also space names. Methods `send` and `receive` must agree in the number and type of the values passed and received, respectively. In this way, passing a space name enables the receiver to be aware of such a space. A data/object space mainly consists of two associative lists, called `subscription` and `collection`. The `subscription` list is a table containing all the objects that have subscribed for a specific type of data/object when it became available in the space. The `collection` list stores all the data/objects available in the space organized by their type. Basic operations on associative lists are `create`, that creates an empty space, `put` that inserts a key-value pair into the list and `get` that retrieves the value associated to a key. When more than one value is associated to the same key, all the values are organized by the order of their arrival.

The programming model provides the means not only to specify and visualize user and agent interactions in a computational context for ubiquitous computing, but also it provides the means to define their formal properties. Among them, reachability and deadlock-free analysis, and other liveness and safeness properties can be established and demonstrated. For example, provided that there are appropriate contextual data for the Hill Climbing problem, then it can be shown that the sketched program in preceding section will find the local maximum. The operational semantics described allows the proof of program correctness. Within specific protocols of nomadic computing, the operational semantics may also proof procedural completeness, but this exposition is outside the scope of this paper.

6 Conclusions

Ubiquitous and nomadic computing open new problems in distributed system design and construction. Among them, visualization of program specification

and execution, in a platform neutral manner are among the most important, as system designers have to deal with the complexities involved in mobile users and agents. From the system designer perspective, instead of introducing context awareness in UML by means of the stereotype mechanism, a Context Aware UML-Sequence visual programming language has been proposed. CA UML-S is being designed to deal with the fundamental problem of human and process mobility in computational contexts that cannot be described with UML diagrams.

References

1. Weiser, M.: Some computer science issues in ubiquitous computing. Commun. ACM 36, 75–84 (1993)
2. Olmedo-Aguirre, J.O., Escobar-Vázquez, K., Alor-Hernández, G., Morales-Luna, G.: ADM: An active deductive XML database system. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 139–148. Springer, Heidelberg (2004)
3. Beguelin, A., Dongarra, J.J.: Graphical development tools for network-based concurrent supercomputing. In: Supercomputing 1991: Proceedings of the 1991 ACM/IEEE conference on Supercomputing, pp. 435–444. ACM, New York (1991)
4. Topol, B., Stasko, J.T., Sunderam, V.S.: Integrating visualization support into distributed computing systems. In: ICDCS, pp. 19–26 (1995)
5. Rivera-de-la-Rosa, M., Olmedo-Aguirre, J.O.: A visual programming environment for eca rules. Advances in Artificial Intelligence and Computer Science. Research on Computing Science 14, 253–264 (2005)
6. Valk, R.: Petri nets as token objects: An introduction to elementary object nets. In: Desel, J., Silva, M. (eds.) ICATPN 1998. LNCS, vol. 1420, pp. 1–25. Springer, Heidelberg (1998)
7. Lomazova, I.A.: Nested Petri nets: Multi-level and recursive systems. Fundam. Inf. 47, 283–293 (2001)
8. de Frutos-Escrig, D., Alonso, O.M., Velardo, F.R.: Ubiquitous systems and Petri nets. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3481, pp. 1156–1166. Springer, Heidelberg (2005)
9. Alhir, S.S.: UML in a nutshell: a desktop quick reference. O'Reilly & Associates, Inc., Sebastopol (1998)
10. Bell, D.: UML's sequence diagram (2004), http://www.ibm.com/developerworks/rational/library/3101.html
11. Hennessy, M., Plotkin, G.D.: Full abstraction for a simple parallel programming language. In: Becvár, J. (ed.) MFCS. LNCS, vol. 74, pp. 108–120. Springer, Heidelberg (1979)

SERS and ANFIS: Fast Identification of the Presence of Retrovirus in CD4 Cells, Cause of AIDS

Jaime De La Torre[1], Francisco Luna[2], Julio Martínez[2], Alejandro Padilla[3], and Miguel Mora[4]

[1] Instituto Tecnológico y de Estudios Superiores de Monterrey. Aguascalientes, Mexico
jtorre@itesm.mx
[2] Instituto Tecnológico de Aguascalientes, Mexico
fjluna@ita.mx, jucemaro@yahoo.com.mx
[3] Laboratorio de Inteligencia Artificial, UA de Aguascalientes, Mexico
apadilla@correo.uaa.mx
[4] Centro Universitario de los lagos, Universidad de Guadalajara, Mexico
mmora@culagos.udg.mx

Abstract. The harmful presence of retrovirus in CD4cells of the human immune system can result in the syndrome of human immunodeficiency known as AIDS, a disease that has extended widely across the entire planet. This paper proposes to obtain characteristic RAMAN spectra with specific peaks detected, eliminating the noise of high frequency (HF) and fluorescence of the signal obtained with SERS and improved with ANFIS. With the spectra cleaned of this noise (HF and fluorescence) the characteristic RAMAN spectra of each microorganism or retrovirus (HIV) in this case is defined. This method provides the specialists with important clinical tools to express an efficient diagnosis of AIDS.

1 Introduction

The existence of several types of cancer [1], multiple factors exist that favor the appearance of cancer in different parts of the human body; in blood, two kinds of cancer appear: a) leukemia caused by the presence and action of a retrovirus (deltaretrovirus), and b) the syndrome of acquired immunodeficiency (AIDS) caused by the presence and action of another retrovirus known as human immunodeficiency virus or HIV (Lentivirus) [2]. HIV affects a type of lymphocytes (leukocytes), called CD4 cells, directly in blood (called "T" cells). HIV/AIDS is one of the most urgent threats to global public health. Some retroviruses cause cancer directly, integrating genes called *oncógenes* in the DNA of the guest cell, thus causing the malignant transformation of normal cells into cancerous cells. Others cause cancer indirectly by activating *proto-oncogen* of the guest. Oncogenic retrovirus has been studied since 1908, [3, 4], Fujinami and Inamoto [5, 6] and it constitutes the most important pathogenic agent of the last decades. The nature of the retroviral genome was discovered in the 1960´s. [7, 8]. At the moment two types of HIV are known; the HIV-1, that causes AIDS [9], and the VIH-II. Both isolated in Institute Pasteur [10, 11, 12]. The number of people living with HIV worldwide in 2007 was estimated at 33.2 million (30.3–36.1 million)

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 936–947, 2008.

[13, 14]. Worldwide, 0.8% (0.7–0.9%) of the adult population (aged 15–49 years) is estimated to be infected with HIV. Sub-Saharan Africa is the region most affected by HIV/AIDS, a total of 22.5 million in 2007. Although other regions are less severely affected, 4 million people in south and south-east Asia, 1.6 million in eastern Europe and central Asia and 1.7 million in Latin America, were living with HIV/AIDS. An estimated 63 000 (49000–98000) people died of AIDS last year in Latin America [13].

All over the world, several types of analyses practiced, generally in blood samples of individuals in study (also other corporal fluids samples are used), determine if an individual is or not infected with the human immunodeficiency virus (HIV) that causes AIDS. Some of these different types of analysis are: Detection of the antibodies and/or antigens of the HIV [15, 16], analysis of viral counts [17–20] and counts of CD4 and CD8 cells [21]. Recently, a new technique has been used to detect virus, Surface-Enhanced Raman Spectroscopy (SERS) [22].

Principles of Surface-Enhanced Raman Spectroscopy (SERS) [22] phenomenon have been applied in multiple scientific areas, like Physics [23], and Chemistry [24]; applications in Biochemistry [25, 26] and specifically in the detection of virus [27, 28]. SERS was used to obtain a Raman spectra of the chains (*A1/long, B1 and A2*) of the syncytial respiratory virus (*SRV*), cause of a type of grippe [27], where this study presents a high index of reproduction, and the time required to obtain results is very short compared to other traditional biochemical analyses mentioned previously. The difference is from hours to seconds. These spectra can be improved using ANFIS (Adaptative Neuro Fuzzy Inference System), a combination of Neuronal Networks (NN) and Fuzzy Logic (FL), which are tools of Artificial intelligence.

ANFIS has been applied in some areas like Architecture [29], in automotive industry [30], in Biochemistry [31], and in Medicine [32–34].

This paper proposes to obtain characteristic RAMAN spectra with specific peaks detected, eliminating the noise of high frequency (HF) and fluorescence of the signal obtained with SERS. With the spectra cleaned off this noise (HF and fluorescence) the characteristic RAMAN spectra of each microorganism or retrovirus (HIV) is defined. It is important to make clear that, at present, we have not conducted experiments to acquire SERS signals because of the lack of the appropriate technology, and that most of our effort has been focused on the usage of artificial intelligence to a) clean up Raman signals –noise removal-, b) recognize Raman Peaks and, c) classify spectra into healthy or infected cells (the latter not reported here for HIV); the results are valid since SERS and non-SE Raman Spectroscopy produce the same kind of spectra, from a signal analysis perspective. The Raman spectra we have used so far are also of the biological type, i.e., spectra from the biopsy of breast cancer intended to detect it. In the sequel, what the work consists of will be explained.

2 Antecedents

2.1 Surface-Enhanced Raman Spectroscopy (SERS) and Virus Detection

Raman Spectroscopy is a high-resolution technique that provides chemical and structural information of organic and inorganic compounds thus allowing its identification. [35]. The Raman spectroscopy analysis is based on the examination of the light

dispersed by a material when affecting a monochrome beam of light. Most of the dispersed light presents the same frequency that the incident light, but a very small fraction displays a change in the frequency, result of the interaction of the light with the matter. The dispersed light that presents frequencies different from the incident radiation is the one that provides information on the molecular composition of the sample and is known as Raman dispersion [35]. A Raman signal is composed by the Raman spectra and two signals considered "noise" inherent to the acquisition of it: the background noise or fluorescence, and the noise of high frequency.

Considering the high index of mutation that virus presents, the efficient detection of these organisms is fundamental to prevent diseases and later epidemics. [28], [22]. With the RAMAN technology improved with the use of silver nanorods, Yiping Zhao et al. from the University of Georgia in the USA have reported that they found the spectra of different types of virus [38], as Adenovirus, Rhinovirus and VIH spectra.

This is an important advance in the field of medicine. A rapid and effective diagnosis of the presence of these microorganisms in the human being will favor the rapid medical prescription and therefore decrease the morbidity and mortality in the population.

2.2 ANFIS Architecture

Adaptative **N**euro **F**uzzy **I**nference **S**ystem (**ANFIS**) is an architecture such as functionally equivalent to diffuse inferential systems (fuzzy), that means, is equivalent to the type of diffuse rules based on Takagi and Sugeno. Some parameters allow establishing the set of training of ANFIS system, and some common rules presented by the fuzzy model of first order by Sugeno are needed [38], [39], [40]. It is important to consider that the ANFIS structure is one of a network with trainable parameters, the reason why the delta rule algorithms and backpropagation of the error is applicable [41]. ANFIS architecture is an interconnected network of a fuzzy system model Takagi-Sugeno of first order, with all its components: fuzzification, inference and defuzzification [41].

3 Materials, Methods and Results

3.1 Removal of Noise Algorithm. Model Proposed. Description of the Algorithm

As it has been mentioned, a Raman signal is composed of the Raman spectra and two inherent signals of noise, the background noise or fluorescence and the noise of high frequency. The adopted strategy is based on determining the presence of possible peaks (maximums with certain morphologic characteristics) in the Raman Signal, isolating the regions (number of waves) in which these peaks happen, and to retain the outside noise of fluorescence in these regions, so that a continuous signal (outlines) is obtained (figure 1). This signal is composed of separated continuous sections with empty segments; the objective of the ANFIS system consists on determining a possible form (by interpolation) of the signal of fluorescence in the empty segments indeed; a sample of this concept is observed in figure 2. The sequence that is followed to remove the noise of the spectra implies the next 7 stages (see figure 3).

Stage 1. Loading Raman Signal. In this block the vector of data containing the Raman signal is introduced; a file containing the spectra data is generated, which is desired to eliminate the noise of HF and fluorescence. Figure 4 shows a Raman signal example to be processed.

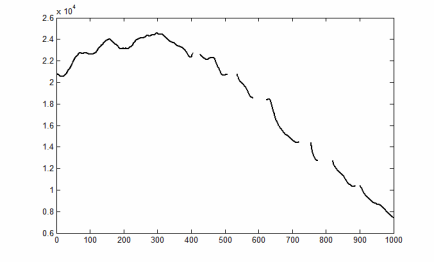


Fig. 1. Discontinuous Raman signals, with the removed regions of peaks

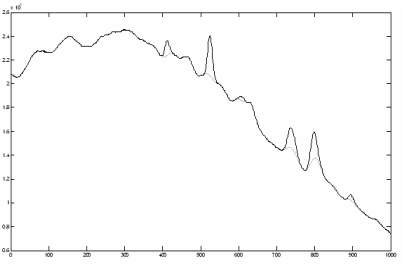


Fig. 2. Discontinuous Raman signal, fluorescence in the removed regions

Stage 2. Normalizing the graph. The idea of making this normalization is to have the signal under the same scale, because the original spectra are in different scales. To reach this objective it was necessary to implement the method of linear interpolation, in x-axis taking 1000 samples (being this an arbitrary value). It is similarly required for the y-axis to be normalized, in this case the scale is from 0 to 1.

This is a better way to work on the spectra, because all the signals are in the same scale and the slope in each Raman signal will be able to be located appropriately and similarly.

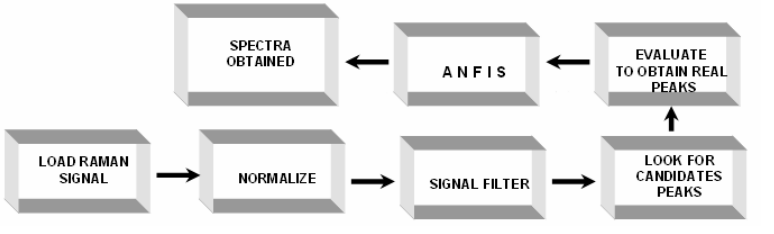


Fig. 3. Sequence of the designed Program to remove the Raman signal's noise

Stage 3. Signal filtering. Filtering is one of the objectives. The type of filter that is used is the "median". With this method, the elimination of the noise of high frequency is obtained. In the representation of the spectra, those small peaks that are found, can cause a bad interpretation of the spectra and therefore generate confusion. Figure 5 shows a graph with HF noise eliminated in all the spectra**.**

Stage 4. Looking for peaks candidates. In order to continue with the spectra analysis, some method that is able to evaluate the resulting signal of the filtration is handled.

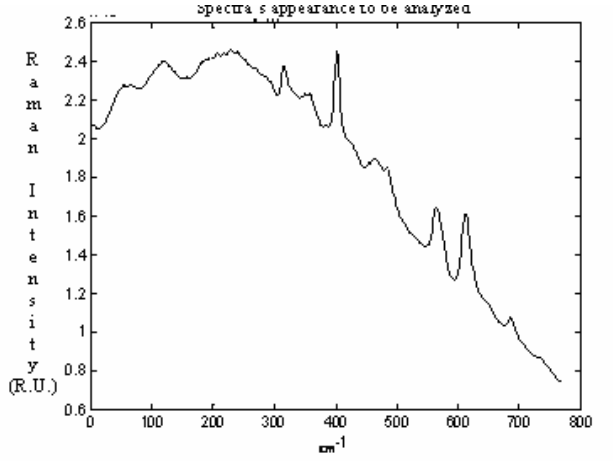


Fig. 4. Original graph obtained by spectroscopic analysis

The way in which this part of the problem is approached is the following: in order to initiate the analysis, a sampling method named "windowing" will be implanted, with the window length equal to 15 samples. By sampling the signal this way, what we find are the maximums within the windows; but in order to set them a limit an evaluation will be done. This evaluation will consider them as Raman peaks if they are centered within that maximum value; if they lean towards a flank they will be discarded. Through this method we can detect the portions of the spectra which may have the shape of a peak, which might later turn to be a characteristic legitimate Raman peak.

Stage 5. Evaluating to obtain real peaks. This stage will be divided in two phases:

(a) Widening of the window. The used methodology to determine whether it is peak or not, is the following; firstly, once all the possible peaks of the signal are obtained, the maximum values acquired are taken for evaluation. Point to point is evaluated descending from the angle that it forms (towards the right side a unit is increased and to the left one a unit is decreased, this is with respect to x-axis), and if the angle is between 60º to –60º, the evaluation continues and it stops when this resulting angle is not within the rank established. In the points in which the condition is not fulfilled, it is stored, and these are considered as the beginning and term of a peak. If a smaller angle were chosen it would lose the aspect of the peak and the errors would appear. In Figure 6 it is possible to observe what is being explained. For this case P1 and P2 are the limits of the peak, that means, the condition of the angle is not fulfilled from it; the little box tries to illustrate the angle that is formed between the points.

As it can be shown in the graph of figure 6, if the beginning and end of a peak do not have the same height a criterion of equalization in these points taking the smallest as reference value (the result will be a peak with the same height in both sides), or taking the original values can be implemented.

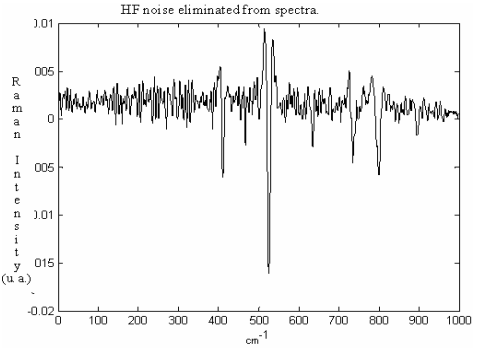


Fig. 5. Graph of the eliminated noise of the input signal

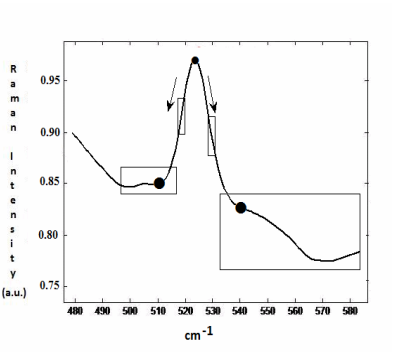


Fig. 6. Evaluation of the Peaks

(b) Determining the Raman peak. The way to know if the peaks found in the first stage are determined as Raman peaks is by means of the *aspect reason (A.R.).* So, the width and height of the possible peak are obtained. These values are obtained from phase 1, and the following relationship is applied: If, A.R. = height/width ≥ 0.6 it is considered a Raman peak, in case of a different value from that relationship it is not a Raman peak. Once the two phases conclude, we have the Raman peaks of the signal, but they still present the problem of the fluorescence (f), which by the use of ANFIS will be eliminated.

Stage 6. ANFIS, developed configuration. ANFIS is used in the union of the points where some peak of the spectra initiate and finish. An adaptive network was constructed, functionally equivalent to the Sugeno´s fuzzy model. The ANFIS Architecture for this Sugeno´s fuzzy model of first order is transparent and effective, Figure 7 shows this network

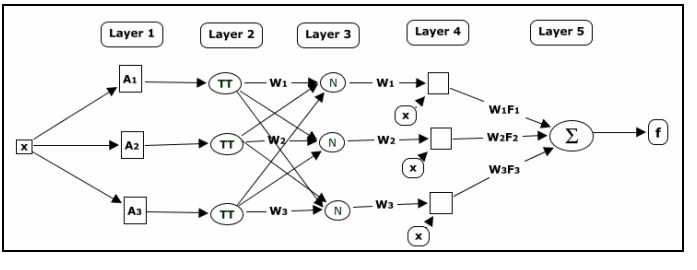


Fig. 7. ANFIS network Structure: One input, three rules, one output

The procedure to apply ANFIS will be explained step by step:

(a) Preparation of data. For explanatory aims, a vector that contains a signal will be known as **y**, or output, and the vector of values of x-axis will be known as **x**, or input. These definitions allow us to establish the set of training of ANFIS system; figure 8 shows an ANFIS scheme with the respective inputs and outputs, as well as the column vectors that form the pairs of data of ANFIS training.

(b) Training of ANFIS. For the training set, a portion of Raman spectra has been selected. The values required for the training are two: the scale in which Raman peak does not appear in the x-axis that goes from 140 to 181 as input variable and the amplitude y as output variable or target that presents the Raman peak. In the mentioned interval a Raman peak exists in the x-axe from 151 to 161. The respective vectors are:

$$\boldsymbol{x} = [140\ 141\ 142\ 143\ 144\ 145\ 146\ 147\ 148\ 149\ 150\ 162\ 163\ 164\ 165\ 166\ 167\ 168\ 169\ 170\ 171\ 172\ 173\ 174\ 175\ 176\ 177\ 178\ 179\ 180\ 181\ ]^{\mathbf{T}}$$

$$\boldsymbol{y} = [\ 0.9636\ \ 0.9650\ \ 0.9663\ \ 0.9675\ 0.9686\ 0.9697\ \ 0.9707\ \ 0.9715\ \ 0.9723\ \ 0.9729\ \ 0.9735\ \ 0.9742\ \ 0.9732\ \ 0.9719\ \ 0.9706\ \ 0.9695\ \ 0.9683\ \ 0.9668\ \ 0.9653\ \ 0.9639\ \ 0.9626\ \ 0.9614\ \ 0.9602\ \ 0.9591\ \ 0.9581\ \ 0.9572\ \ 0.9565\ \ 0.9557\ \ 0.9548\ \ 0.9537\ ]^{\mathbf{T}}$$

The distribution of these values in x, y-axes are shown in Figure 9. Immediately significant results of the process of starting and training of network ANFIS appear.

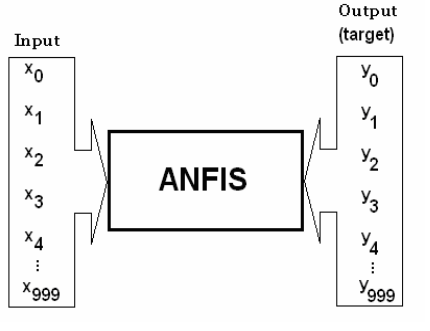


Fig. 8. Training Set for ANFIS

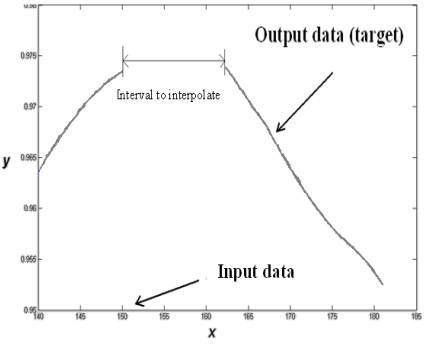


Fig. 9. Training Set for ANFIS

Phase of forward propagation: layer 1, (see figure 7), receives the vector **x** in lot, and calculates the degree of membership $\mu_{Ak}(\boldsymbol{x})$ of each x_i in each membership function A_k; layer 2 (layer Π) makes the fuzzy product of the inputs, but to be only one input, the antecedent is formed by an only condition (if x is…), therefore the output of this layer is equal to that of layer 1; $\boldsymbol{w} = \mu_{Ak}(\boldsymbol{x})$.

In layer 3 every element $w_{i,j}$ is normalized from w to the sum $w_{i,1}+w_{i,2}+w_{i,3}$; Table 2 summarizes the results of the layers process described for the first 10 values of the **x** input signals. In Table 1 column x_i is the individual value of the input; the membership column (μ) and fuzzy product (Π) show the grade of membership of x_i values to each set of inputs; the firing strength or weight of firing of each rule w_i for this particular case is equal to μAi(x), reason why they are not exposed in the table.

Finally, in column $\overline{w}$ the normalized firing strength is found. The following stage of the phase of forward propagation involves layers 4 and 5, implying to solve the (1) linear equations system, in which f_1, f_2 and f_3 are described in the formerly mentioned rules of the fuzzy system at the beginning of the explanation of this stage 6, reason why the equation can be rewritten as a matrix (see equation (2)).

$$\overline{w_1} f_1 + \overline{w_2} f_2 + \overline{w_3} f_3 = f\ . \tag{1}$$

Table 1. Some data of run of propagation of ANFIS for x_1 to x_{10}

		Grade of membership,* Π∗			$\overline{w}$		
I	x_i	$\mu_{A1}(x)$	$\mu_{A2}(x)$	$\mu_{A3}(x)$**	$\overline{w}_1$	$\overline{w}_2$	$\overline{w}_3$ **
1	140	0.0509	0.0000	0.0000	0.9999	0.0001	0.0000
2	141	0.0885	0.0000	0.0000	0.9998	0.0002	0.0000
3	142	0.1453	0.0001	0.0000	0.9996	0.0004	0.0000
6	145	0.4578	0.0011	0.0007	0.9976	0.0024	0.0015
10	149	0.9567	0.0236	0.1502	0.9760	0.0240	0.1532

In which f was replaced by the output (target) T. In order to find P and Q, see that $\theta = [P ; Q]^{\mathbf{T}}$, and then (2) can be solved by the estimator SE, as it follows:

$$\overline{w_1}x + \overline{w_1} + \overline{w_2}x + \overline{w_2} + \overline{w_3}x + \overline{w_3}\begin{bmatrix} p_1 \\ q_1 \\ p_2 \\ q_2 \\ p_3 \\ q_3 \end{bmatrix} = T. \tag{2}$$

$$\hat{\theta} = (A^T A)^{-1} A^T y\,. \tag{3}$$

Continuing with the exemplification, the equation (3) is applied and the values of P and Q are calculated (see Table 2).

Table 2. First estimation of the linear parameters P and Q

Parameter	Value
p_1	0.00098208717961
q_1	0.82678394851996
p_2	-0.00027753790845
q_2	1.01888140570804
p_3	-0.00095894910259
q_3	1.12616038701525

The sum of the square error between the output (target) T, and the output of ANFIS system, is $2.979766593176202e^{-006}$ ($3e^{-6}$); this error is used by the back propagation algorithm as signal of error of the layer from output to the previous one, and with it starts the process of back propagation of the error to fit each trainable element of layer 1 of the ANFIS network; that one is the subject of the following subsection.

In the back propagation phase, the error of approximately $3e^{-6}$ now is the initial error, and it is back propagated to the preceding layers of the ANFIS network, from layer m, a portion of the error is back propagated to layer m_{-1}; thus, from layer 5 to 4 there is an error signal $\varepsilon_{5\text{-}4,}$ from 4 to 3 the signal is $\varepsilon_{4\text{-}3}$, and so on, until reaching error signal $\varepsilon_{2\text{-}1}$.This last one is used to fit the nonlinear trainable elements of layer 1, where

σ and c are the parameters of the fuzzy sets A_k, and the delta rule indicate the amount of adjustment that is necessary to do to these parameters. In general, the update equation (rule delta, and σ is the parameter to be updated) is:

$$\frac{\partial^+ E}{\partial \alpha} = \varepsilon_{m,m-1} \frac{\partial f_{m,m-1}}{\partial \alpha} \quad \text{y} \quad \varepsilon_{m,m-1} = \frac{f_{n+1}}{dx_i} + \sum_{i<j\leq n} \varepsilon_j \frac{\partial f_j}{\partial x_i} . \tag{4}$$

where η is the learning coefficient (in this case, fixed to 0.5), α is the generic parameter to adjust, E is the gradient of the error or vector of the error gradient, that indicates in what sense the adjustment to the parameter α (+ ó -) must be. The signals of error and the adjustments for the memberships functions A_1, A_2 and A_3, are in table 3. Each non linear parameter is fit with the respective increase.

Table 3. Signal of error and increases of the nonlinear parameters

Element	*Value*
$\varepsilon_{2\text{-}1,1}$	0.70130495843930
$\varepsilon_{2\text{-}1,2}$	-0.00009246332232
$\varepsilon_{2\text{-}1,3}$	0.00000000003158
$\Delta\sigma_1$	*5.061675475029995e-006*
$\Delta\sigma_2$	*-3.519487749369613e-013*
$\Delta\sigma_3$	*9.237134301371558e-026*
Δc_1	*-2.074052389768388e-006*
Δc_2	*7.210657827976770e-014*
Δc_3	*3.694058657507537e-020*

3.2 Convergence

The described process in the propagation and back propagation phases continues until certain number of iterations (10000 in this case) and/or until the error diminishes to a specific value.

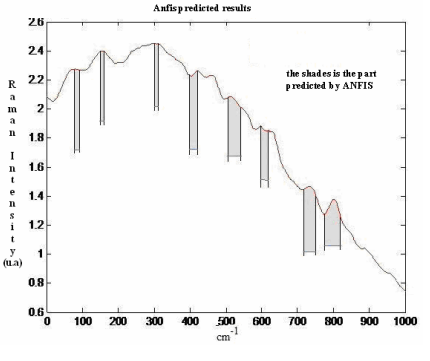


Fig. 10. Final graph of the ANFIS training

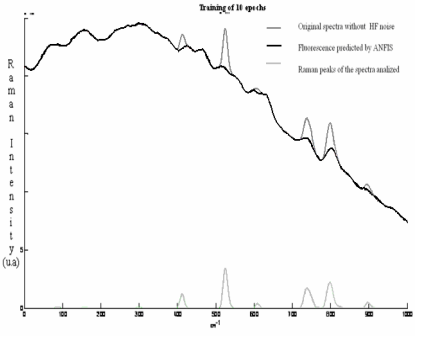


Fig. 11. Graph of final result

(c) Prediction of the fluorescence noise in the regions v with Raman peak. When ANFIS was trained, in figure 10 the prediction of the fluorescence can be observed. The shaded area is the part that is predicted by ANFIS.

Stage 7. Obtaining the spectra. This is the culminating part of the algorithm, in which it must make a joint between the graph of the spectra with noise and the obtained one from ANFIS. With them a subtraction of the signals as it is had in the expression: $E_r = S_r - f$, must be carried out. In figure 11 there are three graphs, the continued one describes the spectra that is obtained to apply the filtration of noise of high frequency, the peaks represent the obtained result of the ANFIS training, finally the peaks below are the graph of the output desired (target) or expected result

4 Conclusions and Future Work

The sequence of the algorithm was described in detail to obtain the graph that contains the Raman peaks of the samples (spectra). It is important to carry out the normalization of the spectra, to obtain the aspect ratio, for the determination of the Raman peaks. And finally with the use of ANFIS the resulting graph appears in the initial scale of the spectra (normalized). The form in which the final spectra will appear is solely the Raman peaks, being these obtained to apply a subtraction between the spectra without noise of HF and those of the ANFIS result. It is important to take into account the computer time for the ANFIS training, the amount of times that it can be trained is infinite, but there exists the amount of times in which the error is reduced significantly or the error is zero. And if not careful computer time will be wrongly invested. On the other hand, to count on patterns of characteristic Raman spectra of microorganisms, as virus, bacteria or fungi, in isolated way or present in organs and cells; provides science with very valuable tools with the purpose of improving the quality of life of the human race. The patterns of retrovirus spectra causes of AIDS (HIV) present in CD4 cells are not yet well defined, but they are in process of being. Future studies in this field will somehow help scientists to find the cure or reach improvements in the health of the people with AIDS, thus improving their life quality.

Raman spectroscopy and ANFIS in the future will be used to characterize microorganism and infected or healthy cells to create new scientific discoveries and to develop new diagnostic tools. This fast method of identification will support the specialists to diagnose AIDS rapidly and at a smaller cost than the traditional methods.

References

1. Tirgan, M.H.: Breast Cancer, A biological imperfection, New York, NY, USA (2006)
2. International Committee of Taxonomy of Virus: ICTV (2007)
3. Ellermann, V., Bang, O.: Experimentelle Leukamie bei Huhnern. Zentralbl. Bakteriol. Parasitenkd. Infectionskr. Hyg. Abt. Orig. 1908, 595–609
4. Rous, P.: A sarcoma of the fowl transmissable by an agent separable from the tumor cells. J. Ex. Med. 13, 397–411 (1911)
5. Vogt, P.K., Coffin, J.M., Hughes, S.H., Varmus, H.E.: Historical introduction to the general properties of retroviruses, pp. 1–25. Cold Spring Harbor Laboratory, New York (1997)

6. Gross, L.: Oncogenic viruses, 3rd edn. Pergamon, Oxford (1980)
7. Duesberg, P.H.: Physical properties of Rous Sarcoma Virus RNA. Proc Natl. Acad. Sci. U S A 60, 1511–1518 (1968)
8. Robinson, W.S., Robinson, H.L., Duesberg, P.H.: Tumor virus RNA's. Proc Natl. Acad. Sci. U S A 58, 825–834 (1968)
9. Organización mundial de la salud OMS: 2006 Report on the global AIDS epidemic, http://www.who.int/research/es
10. Barré-Sinoussi, F., Cherman, J.C., Rey, F., Nugeyre, M.T., Montagnier, L.: Isolation of a T-lymphotropic retrovirus from a patient at risk for AIDS. Science 220(4599), 868–871 (1983)
11. Gallo, R.C., Salahuddin, S.Z., Popovic, M.: Frecuent Detection and Isolation of Cytopathic Retroviruses (HTLV-III) from Patients with AIDS, Risk for AIDS. Science 224, 300–500 (1984)
12. Clavel, F., Guetard, D., Brun-Vezinet, F., et al.: Isolation of a New Human Retrovirus from West African Patients with AIDS. Science 233, 343–346 (1986)
13. Report on the global AIDS epidemic 2008: executive summary. Joint United Nations Programme on HIV/AIDS (UNAIDS). Geneva, UNAIDS (2008), http://www.unaids.org/en
14. WHO report HIV /AIDS estimates are revised downwards 2008: World Health Statistics 2008. Geneva, World Health Organization (2008), http://www.who.int
15. Denis, F., Leonard, G., Sangare, A., et al.: Comparison of 10 Enzyme Immunoassays for Detection of Antibody to Human Immunodeficiency Virus Type 2 in West Africa Sera. J. Clin. Microbiol. 26(5), 1000–4 (1988)
16. de Trousse, B.: BIO-RAD confirmation pour la déetection des anticorps humains anti-VIH1 dans le serum/plasma par immunoblotting, France (2006)
17. Roche, http://www.roche.com
18. Bayer, http://www.bayer.com
19. BioMérieux, http://www.biomerieux.com
20. Booklet Abbott AXSYM System VIH $^{1/2}$gO, Germany(2006), http://www.abbott.com
21. InfoRed SIDA. Proyecto del Centro para la Educación y Entrenamiento sobre SIDA de Nuevo México, http://www.aidsinfonet.org
22. Shanmukh, S., Jones, L., Driskell, J., Zhao, Y., Dluhy, R., Tripp, R.A.: Rapid and Sensitive Detection of Respiratory Virus Molecular Signatures Using a Silver Nanorod Array SERS Substrate. Nano Lett. 6(11), 2630–2636 (2006)
23. Young, M.A., Stuart, D.A., Lyandres, O., Glucksberg, M.R., Van Duyne, R.P.: Surface-enhanced Raman spectroscopy with a laser pointer light source and miniature spectrometer1. Canadian Journal of Chemistry 82(10), 1435–1441 (2004)
24. Brolo, A.G., A.: Surface-enhanced Raman scattering (SERS) from a silver electrode modified with oxazine 720(1). Canadian Journal of Chemistry 82(10), 1474–1480 (2004)
25. Cînta Pînzaru, S., Leopold, N., Pavel, I., Kiefer, W.: Raman, SERS and theoretical studies of papaverine hydrochloride and its neutral species. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy 60(8-9), 2021–2028 (2004)
26. Cînta Pînzaru, S., a Leopolda, N., Kiefer, W.: Vibrational spectroscopy of betulinic acid HIV inhibitor and of its birch bark natural source. Talanta 57(4), 625–631 (2002)
27. Shanmukh, S., Jones, L., Zhao, Y.P., Driskell, J.D., Tripp, R.A., Dluhy, R.A.: Identification and classification of respiratory syncytial virus (RSV) strains by surface-enhanced Raman spectroscopy and multivariate statistical techniques. Analytical and Bioanalytical Chemistry 390(6), 1551–1555 (2008)

28. Kincade, K.: SERS and silver nanorods quickly reveal viral structures Laser Focus World. Career and Technical Education 43(1), 49 (2007)
29. Ayata, A.: Adaptive neuro-fuzzy inference systems (ANFIS) application to investigate potential use of natural ventilation in new building designs in Turkey. Energy Conversion and Management 48(5), 1472–1479 (2007)
30. Zarandi, M.H.F., Turksen, I.B., Maadani, B.: Customer satisfaction assessment with fuzzy queries and ANFIS for an automotive industry. Fuzzy Information. In: Processing NAFIPS, IEEE Annual Meeting, 27-30 June 2004, vol. 2, pp. 723–728 (2004)
31. Buyukbingol, E., Sisman, A., Akyildiz, M., Alparslan, F.N., Adejare, A.: Adaptive neuro-fuzzy inference system (ANFIS): a new approach to predictive modeling in QSAR applications: a study of neuro-fuzzy modeling of PCP-based NMDA receptor antagonists. Bioorganic & medicinal chemistry 15(12), 4265–4282 (2007)
32. Subasi, A.: Application of adaptive neuro-fuzzy inference system for epileptic seizure detection using wavelet feature extraction. Computers in Biology and Medicine 37(2), 227–244
33. Derya, E.: Adaptive neuro-fuzzy inference system employing wavelet coefficients for detection of ophthalmic arterial disorders, vol. 34(3), pp. 2201–2209 (April 2008)
34. Heejun, S., Seongu, L., Dongwon, K., Gwitae, P.: New methodology of computer aided diagnostic system on breast cancer. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3498, pp. 780–789. Springer, Heidelberg (2005)
35. Raman, C.V.: Nature 108, 367 (1921)
36. Raman, C.V., Krishnan, K.S.: A new type of Secondary Radiation. Nature 121, 619 (1928)
37. Contreras, J.G.: Espectroscopia RAMAN y estructura molecular. UNECO 1 (1987)
38. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. Transactions on Systems Man and Cybernetics 15(1), 116–132 (1985)
39. Sugeno, M., Kang, G.T.: Structure identification of fuzzy model. Fuzzy sets and Systems 28, 15–33 (1988)
40. Takagi, T., Sugeno, M.: Derivation of fuzzy control rules from human operator´s control actions. In: Proceedings of the IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, pp. 55–60 (July 1983)
41. Roger, J.S.: Neuro-fuzzy and Soft Computing. Prentice Hall, Nj (1997)

A Radon Transform Based Approach for Extraction of Blood Vessels in Conjunctival Images

Reza Pourreza[1], Touka Banaee[2], Hamidreza Pourreza[3], and Ramin Daneshvar Kakhki[2]

[1] Electrical Engineering Department, Ferdowsi University of Mashhad, Iran
reza_pourreza@yahoo.com
[2] Ophthalmic Research Center, Khatam-Al-Anbia Hospital,
Medical Sciences University of Mashhad, Iran
{banaeet,DaneshvarR}@mums.ac.ir
[3] Computer Engineering Department, Ferdowsi University of Mashhad, Iran
hpourreza@um.ac.ir

Abstract. This paper proposes a local Radon transform-based algorithm for extraction of blood vessels in conjunctival images. This algorithm divides the image into overlapping windows and applies Radon transform to each window. Vessel direction in each window is found by detection of peak in Radon space. The proposed algorithm is capable of extracting blood vessels with a variety of widths. According to vessel width, extracted blood vessels are classified into some predefined classes and several statistics are computed for each class. Since the Radon transform is robust against noise, proposed algorithm is noise-independent and is more robust in comparison with other available algorithms.

1 Introduction

Eye is the only organ in the living subject where small vessels are directly visible, either in the conjunctiva, or in the retina. Diabetic retinopathy is a well known microvascular complication of diabetes. There is a standardized classification system for diabetic retinopathy and much work has been done in this field [1,2]. But there are few studies about disturbances of conjunctival vessels in diabetes and other diseases [3]. This is due to the great variability of the shape and course of conjunctival vessels. To detect any changes in the conjunctival vessels, there is a need for an objective measure. Clinical classification of conjunctival vessels is a difficult and observer-dependent task which may not be reproducible. This led us to think of vessel detection techniques in this regard.

Many researchers have invested in the vessel detection field; however, most works focus on retinal images [4,5]. The vessel detection techniques fall into four main categories: local operators, matched filters, vessel tracking and neural networks [6]. In this paper a Radon transform-based algorithm is proposed. The Radon transform is able to transform line-containing images into a domain where each line in the image gives a peak or a valley in that domain. The Radon transform is less sensitive to noise in the image than other line detectors, because the intensity fluctuations due to noise tend to be cancelled out by the process of integration [7]. Radon transform makes our algorithm more robust and less sensitive to noise than other proposed algorithms. In

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 948 – 956, 2008.

addition, the utilized technique for vessel detection simplifies the statistical analysis of the input conjunctiva image.

In the proposed algorithm conjunctiva image is firstly partitioned into some overlapping blocks. Local Radon transform is then applied to each block. The peak of sub-image in Radon space would be associated with the sub-vessel (if a sub-vessel lays in the sub-image). The projection angle which includes the peak is further analyzed for vessel presence validation and vessel width calculation. The extracted sub-vessels are then refined and the blocks are merged. The algorithm outputs some masks for different vessel widths. Each mask is associated with a special vessel width and is a map for those vessels with that specific width. The masks are analyzed for vessel classification and statistical analysis. The extracted statistical information is useful for medical purposes.

The rest of paper is organized in 6 parts as follow. Section 2 is *Image Partitioning* and explains the utilized partitioning method. Section 3 is *Local Radon Transform* which is concerned about local Radon transform and image analysis in Radon space. Section 4 is *Vessel Validation* and deals with validation of extracted sub-vessels and also width calculation of valid sub-vessels. Section 5 is *Vessel Refinement* and explains the method that extracted sub-vessels are compared with real ones to acquire better results. Section 6 is *Statistical Analysis* and in this section the output masks analysis for extracting statistical information, is discussed. The final section is *Results and Discussion*.

2 Image Partitioning

In order to extract blood vessels, sub-vessels should be extracted in local windows. The window size (n) has a direct effect on the extraction accuracy. A small/large n would lead to extract thin/thick vessels accurately while thick/thin vessels would not be extracted. A good combination of thin and thick vessels can be extracted by adjusting the n value. Another important parameter which affects the algorithm's accuracy is the windows overlapping, because non-overlapping windows extremely limit the quality of detected vessels. Thus in the proposed system a parameter $step$ is used which defines the adjacent windows overlapping ratio. Fig. 1 depicts the relation of $step$ and window size (n).

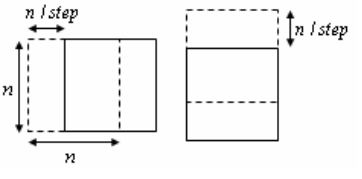


Fig. 1. Window size vs. Overlapping ratio

3 Local Radon Transform

The available conjunctiva images are mostly red-free, i.e. their R component in RGB color space equals zero. In the proposed algorithm, G component of sub-images are

selected for applying local Radon transform. The reason is the high contrast between blood vessels and image background in G component. G component is then complemented; this makes the vessels to be associated with peaks in Radon space (not valleys). In spite of the high contrast, we enhance the quality of sub-images before applying radon transform. Image enhancement i.e. histogram equalization improves the sub-image contrast and causes more discrimination between sub-vessels and conjunctiva's background and increases the total accuracy.

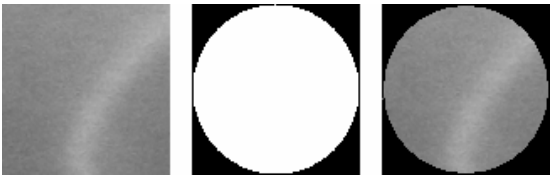
Local Radon transform is applied to the enhanced sub-image in the next step. The Radon transform of a continuous function $g(x, y)$ in 2-D Euclidian space is defined by (1)

$$R(\rho,\theta) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x, y)\delta(\rho - x\cos\theta - y\sin\theta)dxdy \tag{1}$$

where $\delta(x)$ is Dirac function. When $g(x, y)$ is an image, the integral boundaries would change to image dimensions as (2)

$$R(\rho,\theta) = \int_{0}^{Y}\int_{0}^{X} g(x, y)\delta(\rho - x\cos\theta - y\sin\theta)dxdy \tag{2}$$

where in our case both X and Y equal n. According to (1) and (2) the amplitude of projection in diagonal directions ($\theta = 45^{\circ}, \theta = 135^{\circ}, \rho = n\sqrt{2}$) is higher than other directions, thus the peak of Radon transform is more likely to happen in diagonal directions. To eliminate the diagonal effect, the input sub-image is firstly masked using a circle. The masking process is shown in fig. 2.



(a) (b) (c)

Fig. 2. Masking Process, a) sub-image, b) mask, c) masked sub-image

Then local Radon transform is applied to the masked image. As mentioned before, sub-vessels in sub-images are associated with peaks in Radon space; therefore at this stage peaks should be detected. However, the noise in the image moves the place of peaks or makes some fake peaks that would lead to false vessel detection. In order to remove the noise effect, the sub-image is filtered in radon space using a 3×3 mean filter. The peak and the corresponding projection angle are detected in the filtered Radon transform. The profile in which peak occurs, is a candidate that might contain a sub-vessel. This profile is further analyzed for validation of candidate sub-vessel.

4 Vessel Validation

An easy solution to the vessel validation problem is to compare the peak amplitude with a predefined threshold. However, this is a background-dependent solution, because the more/less bright background, the higher/lower peak amplitude. In our algorithm, effect of background and blood vessel on the peak amplitude are differentiated and then the comparison with the threshold is performed. In order to differentiate the background and vessel effects on peak amplitude, mean of sub-image is multiplied by window size (n) and then is subtracted from the profile as shown in (3)

$$P(i) = P(i) - n \times mean(subimage),\ i = 1,...,M \tag{3}$$

where P denotes profile and M is the number of bins in the profile.

Since the sub-image's mean is a good indicator of background, the multiplication of mean and window size would be the effect of background on the profile. The subtracted profile is background-free and in order to validate the presence of a sub-vessel in the candidate projection angle, profile's peak is normalized by division to the window size (n) and then compared with the threshold.

If the peak amplitude is bigger than threshold, the detected sub-vessel is verified and the algorithm should calculate sub-vessel's width. For this purpose the normalized profile is utilized. The normalized profile is the result of dividing the subtracted profile by window size. Fig. 3 depicts a normalized profile.

In fig. 3, i_p is the peak's index and is associated with the sub-vessel's centerline. The interval $[i_{\min}, i_{\max}]$ around i_p is the sub-vessel's area and $i_{\max} - i_{\min} + 1$ is using (4) to (7). In (4) to (7), α is a constant that equals 0.7 and P denotes profile.

$$i_{\max} > i_{\min} \tag{4}$$

$$P(i_{\max}) = \alpha \times P(i_p) \tag{5}$$

$$P(i_{\min}) = \alpha \times P(i_p) \tag{6}$$

$$w = i_{\max} - i_{\min} + 1 \tag{7}$$

Giving $i_{\min}$, $i_{\max}$, projection angle and window size, a mask is prepared which presents coarse sub-vessel's coverage area in the sub-image. For this purpose, on a black $n \times n$ block a white line is drawn which its angle equals projection age and its position is determined by $i_{\min}$ and $i_{\max}$ (so the width of drawn line equals w). A sample of local vessel mask is shown in fig. 4.

The local vessel mask is processed in the following sections for vessel map extraction and statistical analysis.

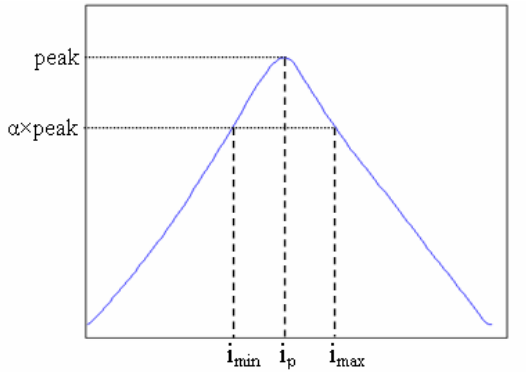


Fig. 3. Normalized Peak-Containing Profile

Fig. 4. Local Vessel Mask

5 Vessel Refinement

The output local vessel mask of *vessel validation* section represents sub-vessel which lays in the sub-image. However, it is not a good representation, because the drawn white line and the real sub-vessel may not fully cover each other and this phenomena decreases the quality of final detected vessel map and also disturbs the extracted statistical information. To improve the quality of extracted sub-vessels in the sub-image, the local vessel mask should be compared with real sub-vessels. For this purpose, the input sub-image is converted to a binary image which white pixels represent vessel area and black pixels represent background. A binarized sub-image is shown in fig. 5.

Fig. 5. Binary Sub-image

In order to determine the threshold for image binarization, the local vessel mask is compared with input sub-image and two gray level means are computed. First gray level mean (m_1) is the mean of those pixels' gray level in sub-image which their corresponding pixels in local vessel mask lay on white line. Second gray level mean (m_2) is the mean of those pixels' gray level in sub-image which their corresponding pixels in local vessel mask lay on black background. The binarization threshold equals mean of m_1 and m_2.

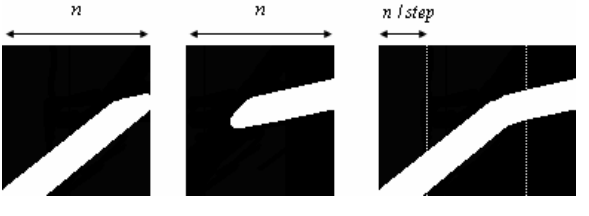
Essentially all white pixels in the acquired binary image are not associated with vessel pixels because image binarization is a noise-sensitive process. In our algorithm, the local vessel mask is compared with the binary sub-image to achieve a fine representation of sub-vessels and the result is named fine local mask. The comparison process is done through a logical AND as shown in (8).

$$FLM(i,j) = LVM(i,j)\, AND\, BSI(i,j) \qquad i,j = 1,...,n \tag{8}$$

where FLM , LVM and BSI denote fine local mask, local vessel mask and binary sub-image respectively. Fig. 6 shows the fine local mask.

Fig. 6. Fine Local Mask

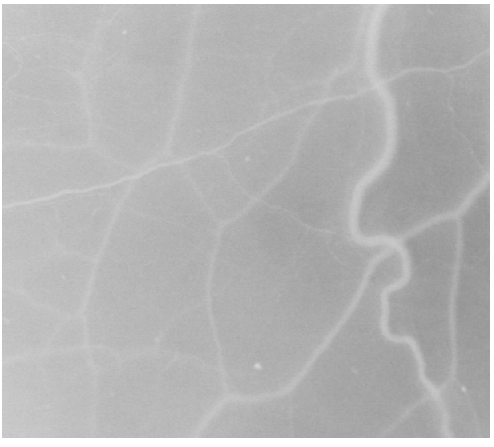
So far, the input image was partitioned into sub-images; a sub-vessel was extracted from each sub-image (if there exist) and the output of algorithm for each sub-vessel was its fine representation (fine local mask) as well as its width (w). At last, fine local masks are merged to obtain the vessel map of the input image. In order to merge fine local masks, adjacent masks are put beside each other considering the overlapping ratio; on the overlapping region a logical OR is performed. The merging of two adjacent masks is shown in fig. 7.



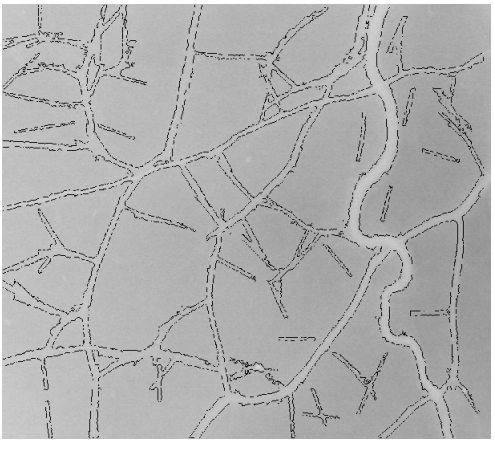
(a) (b) (c)

Fig. 7. Merging Process of adjacent masks, a) mask1, b) mask2, c) merged mask

Fig. 8 depicts a conjunctiva image which its vessel map is overlaid.



(a)



(b)

Fig. 8. Overlaid Vessel Map, a) G component of input image, b) overlaid detected vessels

Our algorithm is capable of extracting vessels whose widths vary over $[1, n-2]$ range. The merging process is done for each vessel width i.e. for each vessel width a vessel map is provided that contains those vessels with that specific width. Thus, $n-1$ vessel maps are prepared; one is the overall vessel map (m) and the others are associated with $n-2$ different vessel widths (m_i, $i = 1,...,n-2$ where i denotes vessel width,). The $n-1$ vessel maps are output to *statistical analysis* section.

6 Statistical Analysis

In this section, statistical information about the input conjunctiva image is extracted from the $n-1$ vessel maps. Statistical information includes: vessel width mean, vessel width variance and vessel density. All masks dimensions equal $H \times W$ where H and W denote image height and width respectively. In order to extract statistical information, the following $n-1$ statistics are computed using (9)

$$p = \frac{1}{H \times W} \sum_{j=1}^{H} \sum_{k=1}^{W} s \times m(j,k)$$
$$p_i = \frac{1}{H \times W} \sum_{j=1}^{H} \sum_{k=1}^{W} s \times m_i(j,k), \; i = 1,...,n-2 \tag{9}$$

where m and m_i are defined in *Vessel Refinement* section, s is the scale factor in $\mu m / pixel$ and p denotes the number of white pixels (vessel pixels) to all pixels ratio in the corresponding mask.

Vessel density (d), which represents the density of vessels in the input image, equals p. Vessel width mean (μ) and variance (σ^2) are evaluated using (10) and (11) respectively.

$$\mu = \sum_{i=1}^{n-2} s \times i \times p_i \tag{10}$$

$$\sigma^2 = \sum_{i=1}^{n-2} (s \times i - \mu)^2 p_i \tag{11}$$

7 Results and Discussion

We applied our algorithm to 80 conjunctiva images. The input images dimensions were 768×864, window size (n) was 65, *step* was 3 and validation threshold was chosen 8.5. The extracted blood vessels were checked by a physician and the algorithm was evaluated subjectively. The acquired results demonstrate the high accuracy of our algorithm in vessel extraction and robustness against noise. As can be seen in fig. 8.b (which shows a sample result), our algorithm extracts both thin and thick vessels well even in low contrast regions.

References

1. Fowler, M.J.: Microvascular and Macrovascular Complications of Diabetes. Clinical Diabetes 26(2), 77–82 (2008)
2. Klein, R., Klein, B.E., Moss, S.E., Wong, T.Y.: Retinal vessel caliber and microvascular and macrovascular disease in type 2 diabetes: XXI: the Wisconsin Epidemiologic Study of Diabetic Retinopathy. Ophthalmology 114(10), 1884–1892 (2007)

3. Owen, C.G., Newsom, R.S.B., FRCOphth, Rudnicka, A.R., Ellis, T.J., Woodward, E.G.: Vascular Response of the Bulbar Conjunctiva to Diabetes and Elevated Blood Pressure. Ophthalmologhy 112(10), 1801–1808 (2005)
4. Kirbas, C., Quek, F.: A Review of Vessel Extraction Techniques and Algorithms. ACM Computing Surveys 36(2), 81–121 (2004)
5. Sang, N., Tang, Q., Liu, X., Weng, W.: Multiscale Centerline Extraction of Angiogram Vessels Using Gabor Filters. In: Zhang, J., He, J.-H., Fu, Y. (eds.) CIS 2004. LNCS, vol. 3314, pp. 570–575. Springer, Heidelberg (2004)
6. Estabridis, K., Defigueiredo, R.: Blood Vessel Detection via a Multi-window Parameter Transform. In: Proc. of CBMS 5960 (2006)
7. Zhang, Q., Couloigner, I.: Accurate Centerline Detection and Line Width Estimation of Thick Lines Using the Radon Transform. IEEE Transaction on Image Processing 24, 310–316 (2006)

Identification and Quantification of Pulmonary Emphysema through Pseudocolors

John Hebert da Silva Felix[1], Paulo César Cortez[1], Pedro Pedrosa Rebouças Filho[1], Auzuir Ripardo de Alexandria[2], Rodrigo Carvalho Sousa Costa[1], and Marcelo Alcantara Holanda[3]

[1] Federal University of Ceará - UFC, Department of Teleinformatic Engineering - DETI, Fortaleza, Brazil
johht@yahoo.com.br,pedrosa_rf@hotmail.com,{cortez,rodcosta}@deti.ufc.br
[2] Federal Technologic Educational Center of Ceará - CEFET, NSMAT, Fortaleza, Brazil
auzuir@gmail.com
[3] Walter Cantídio University Hospital of the Federal University of Ceará, Fortaleza, Brazil
marceloalcantara2@gmail.com

Abstract. Chronic Obstructive Pulmonary Disease (COPD) is a world health problem with high morbidity and mortality. High-Resolution Computed Tomography (HRCT), is an excellent tool for early detection of emphysema component of COPD. Despite this fact, HRCT presents limitations inherent to the subjective analysis of the gray scale image that directly compromises the accuracy for both diagnosis and precise determination of the disease extension. The objective of this paper is present a colored mask algorithm (CMA) to identify and quantify the emphysema, enhancing its visualization through pseudocolors. We studied 21 images of 7 patients with COPD and 1 healthy volunteer. The CMA applies colors to the segmented lungs according to pre-defined ranges of Hounsfield units. CMA automatically calculates the relative area occupied by tomographic densities within the pre-defined ranges, allowing precise quantification of diseased and normal parenchyma. Future works are needed in order to validate the incorporation of the CMA in the image assessment of emphysema in COPD patients.

1 Introduction

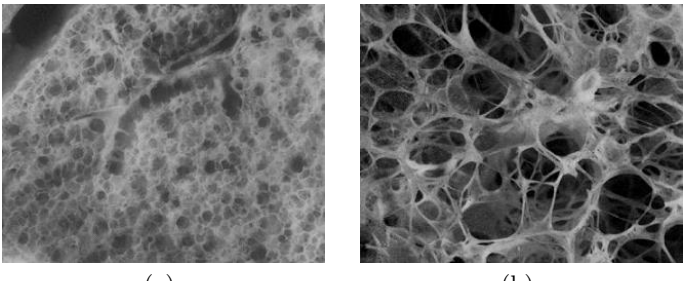
Chronic Obstructive Pulmonary Disease (COPD) is a world health problem with high mortality and smoking is this main cause. World Health Organization (WHO) estimates that the global population of smokers will increase by 1.2 to 1.6 billion between the years 2000 to 2030 [1,2]. Examples of lung tissues normal and with emphysema are shown in Fig. 1

COPD is a preventable and treatable disease characterized by airflow limitation that is not fully reversible and emphysema is main a component [3]. Emphysema is defined as an abnormal, permanent enlargement of the air spaces

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 957–964, 2008.

distal to the terminal bronchioles, accompanied by the destruction of their walls, without evident fibrosis [4].

COPD is usually classified into four stages according to symptoms and lung function [5]. Because emphysema is characterized by a decrease in tomographic density (in Hounsfield units, HU) attributable to the combination of loss of lung tissue, decreased pulmonary blood flow and increased areas of air trapping, much work has been done to measure the extent of emphysema with computed tomography (CT). Despite this fact, lung densitometry has not yet been applied for early detection or quantification of emphysema. One reason for this is that CT has primarily been developed for visual interpretation of images in gray scale and has not been standardized for quantitative measurements or colored images related to lung densities [6]. Therefore, there is no consensus on the evaluation of COPD severity by high-resolution technique (HRCT) images, mainly because precise quantitative date are not routinely obtained.



(a) (b)

Fig. 1. Lung tissue (a) normal and (b) emphysema [7]

Computed tomography, especially the HRCT, is an excellent tool for early detection of the emphysema component in COPD [8,9,10]. A computer vision system able to detect, quantify and better visualize the components of abnormal hyperaeration found in emphysematous lungs in HRCT images may aid in the early and precise diagnosis of the presence and extension of emphysema in COPD patients [11].

The identification of emphysema of mild degree is particularly, a very common problem. In this case the emphysema usually may not be identified, due to limitation of the human vision perception [12]. An image processing method for the detection and quantification of emphysema may increase the sensitivity of HRCT in the diagnosis of emphysema, even in very early stages.

The objective of this paper is to propose a colored mask algorithm to identify and quantify the emphysema component in HRCT images of COPD patients.

This paper is organized as follows: Section 2 describes the acquisition of images and the concepts of pseudocolor segmentation. In Section 3 the results are shown with CMA and in Section 4 the conclusions and perspectives of future works are discussed.

2 Materials and Methods

2.1 Study Subjects

All patients (seven) had the diagnosis of COPD according to the Global Initiative on Obstructive Lung Disease (GOLD) [5]. One healthy volunteer was also studied for comparison.

2.2 Acquisition of HRCT Images

HRCT was done by an AUKELET TOSHIBA TSX-003A (TOSHIBA MEDICAL, TOKIO, JAPAN) CT scan. Before obtaining the images the scan was calibrated for air (-1000 HU) and for water (0 HU) densities with an appropriate phantom. All images were photographed in a window width of 1600 HU and a window level of -600 HU. Technical parameters were: 1,5 mm collimation, 120 Kv, 200mA, matrix of 512 x 512, field of view (FOV) of 312mm and 1,8s scanning time. The lungs were scanned at three regions: apex (2 cm above the aortic arc), hilum (1cm below the carina) and basis (2 cm above the diaphragmatic dome). To assess the efficiency of the CMA in a different image acquisition process, a CT image of a healthy volunteer acquired from another CT scan equipment was also evaluated. This image of a healthy volunteer was acquired without filter of HRCT.

2.3 Pseudocolor Segmentation

The processing of images through pseudocolor (false color) consists of assigning colors to gray values based on a specified criterion. The principal motivation of this method is the fact that humans can discern thousands of color shades and intensities compared to only two dozen or so shades of gray. The pseudocolor method allows a human viewer to recognize certain areas and/or the details in the images [13].

The intensity (density) slicing is one of the simplest examples of pseudocolor processing, the method can be viewed as one of placing planes parallel to the coordinate plane of the image. Interpreted an image as a 3D function, as shown in Fig. 2, the function $f(x, y) = l_i$ is used to define the slicing planes, where l_i corresponds to the gray level intensity of the plane. The levels beetween two planes is assigned to one specified color [13].

Gray-level to color assignments are made according to the relationship:

$$f(x, y) = c_i \qquad if\ f(x, y)\ \epsilon\ V_i, \tag{1}$$

where c_i is the color associated with the i_{th} range of intensity levels of gray V_i, defined by the partitioning planes at $l = i - 1\ and\ l = i$.

2.4 The Color Mask Algorithm

With the colored masks (CMA) the image is sliced using various slicing planes, according to pre-defined ranges of lung densities in HU.

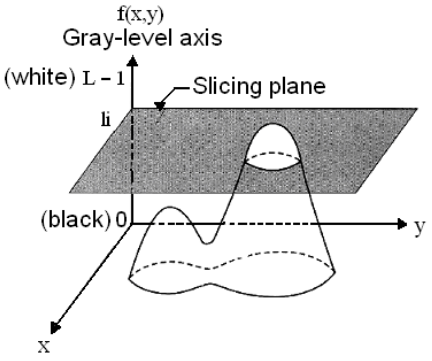


Fig. 2. Geometric interpretation of the density-slicing technique [13]

The pre-defined ranges of lung densities are shown in Table 1. The CMA applies colors to the pixels in the segmented lungs according to pre-defined ranges of HU such as regions of emphysematous areas (< -950 HU), normally aerated tissue (-950 to -500 HU), poorly aerated lungs (-500 to -100) and blood vessels or nonaerated parenchyma (-100 to +100 HU). Furthermore, the CMA automatically calculates the relative area of the lungs occupied by pixels with tomographic densities within the pre-defined ranges, in percentage, allowing precise quantification of diseased parenchyma [14,15].

Table 1. Ranges of lung densities according to the degree of aeration and the color applied

Regions	Ranges of lung densities	Colors applied
hyperaerated	-1000 to -950 HU	light green
normally aerated	-950 to -500 HU	blue
poorly aerated	-500 to -100 HU	magenta
nonaerated	-100 to +100 HU	dark red

3 Results

We studied twenty one images of 7 patients with COPD and of 1 healthy volunteer. Representative images of a COPD patient and of the healthy volunteer are shown in Fig. 3 and 4, respectively.

Initially the images were automatically segmented using the method proposed by Felix *et al.* [16]. This was necessary to extract the lung regions from HRCT images. After the segmentation, the image is processed using the CMA and then, the identification and quantification of lung densities is done automatically.

As seen in Fig. 3 the contrast of lung structures is very low in gray scale and much better after the CMA processing enhancing the visualization of hyperaerated areas, emphysema.

The CMA can apply many pseudocolors in various pre-defined ranges of lung densities. The CMA processing with the pseudocolors highlights the contrast of lung structures as shown in in Fig. 3 and 4.

Table 2. Results obtained by the CMA for 7 patients with COPD and an image of a healthy volunteer

Images	Cuts	Hyper Aerated	Normally Aerated	Poorly Aerated	Nonaerated	Total
Patient 1	Apex	67.30%	29.08%	2.21%	1.13%	99.72%
	Bases	36.30%	59.16%	2.62%	1.70%	99.78%
	Hilum	58.63%	38.73%	1.41%	0.99%	99.76%
Patient 2	Apex	33.77%	62.07%	2.80%	1.04%	99.68%
	Bases	54.09%	42.86%	2.10%	0.80%	99.85%
	Hilum	40.55%	52.92%	3.10%	2.98%	99.55%
Patient 3	Apex	9.42%	78.94%	8.63%	2.65%	99.64%
	Bases	45.63%	49.40%	3.24%	1.47%	99.74%
	Hilum	15.36%	74.33%	6.08%	4.00%	99.77%
Patient 4	Apex	33.82%	60.34%	3.45%	1.68%	99.29%
	Bases	21.63%	71.57%	4.41%	1.98%	99.59%
	Hilum	30.79%	62.60%	3.37%	2.90%	99.66%
Patient 5	Apex	32.75%	63.13%	2.66%	1.10%	99.64%
	Bases	22.37%	69.55%	4.80%	3.06%	99.78%
	Hilum	14.47%	81.48%	3.13%	0.90%	99.98%
Patient 6	Apex	11.81%	78.11%	7.75%	1.90%	99.57%
	Bases	11.07%	81.94%	4.47%	2.31%	99.79%
	Hilum	9.44%	83.49%	4.10%	2.90%	99.93%
Patient 7	Apex	23.15%	71.61%	3.41%	1.30%	99.47%
	Bases	17.93%	76.54%	3.40%	1.98%	99.85%
	Hilum	19.34%	74.48%	3.41%	2.56%	99.79%
Voluntary	Bases	0.35%	97.44%	1.90%	0.31%	100.00%

The CMA calculation of the percentage of the relative area occupied by the pixels within the pre-defined densities ranges for the 7 patients and the healthy volunteer are shown in Table 2. The total sum does not reach 100% due to noise inherent to the application of the HRCT reconstruction algorithm (filter), at the time of CT images acquisition.

The small differences between the results obtained from COPD patients occurred because of noise introduced by the HRCT tomography equipment. There was no noise in the image of the healthy volunteer as expected considering that no filter was used in this case.

Analyzing the quantification of relative areas of emphysema in Table 2, we can observe that patient 1 may be classified as having severe emphysema, mainly in upper lobes (apex) reaching 67.3% of the lungs, while patients 4 and 6 have

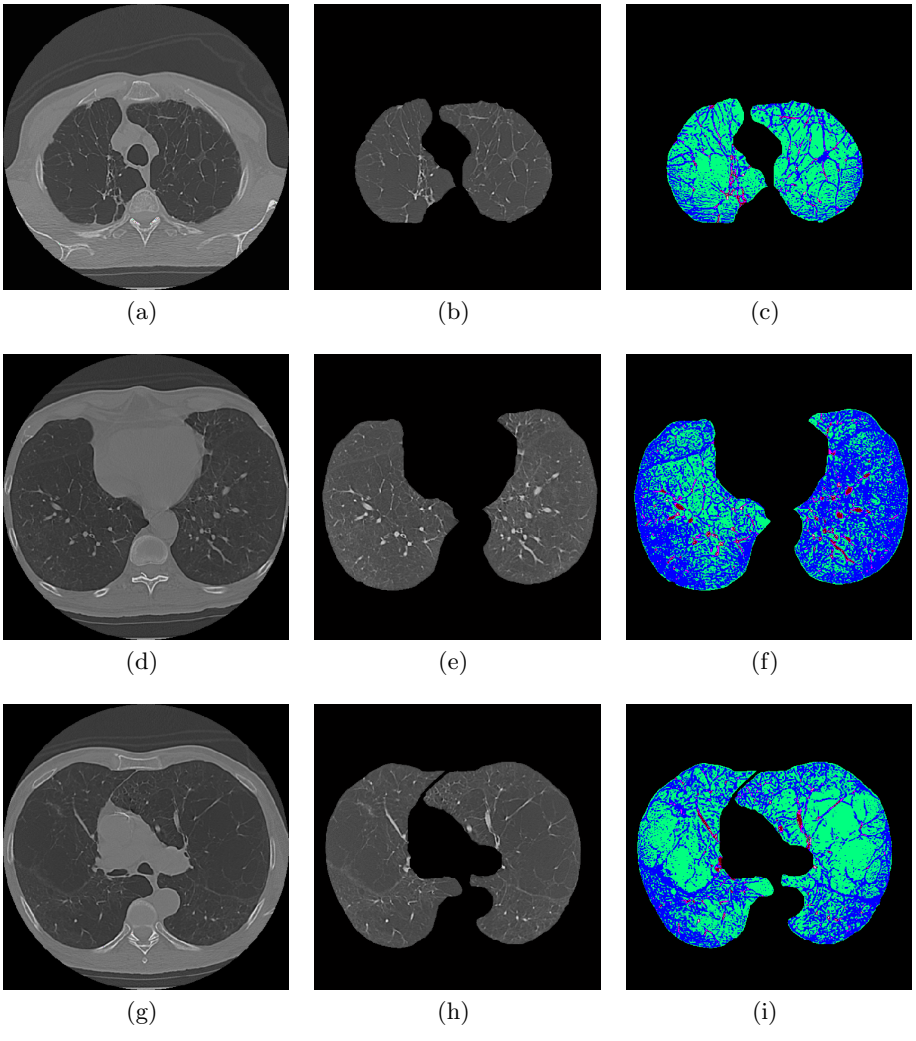
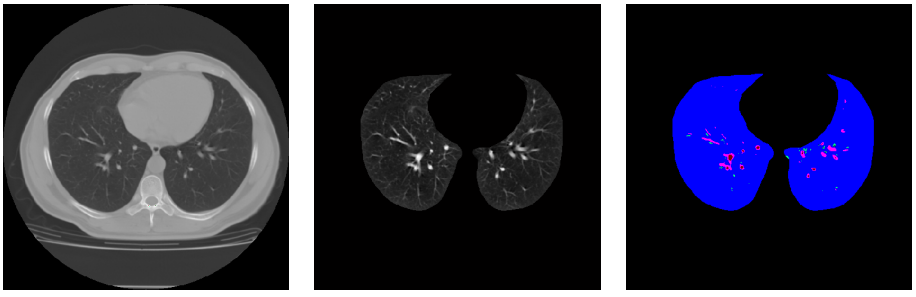


Fig. 3. Representative HRCT images of a patient with COPD. On the left column are shown the original HRCT images. In the center column segmented images with the algorithm proposed by [16] and in the right column the results obtained with the CMA to: hyperaerated (-1000 to -950 HU, light green), normally aerated (-950 to -500 HU, in blue), poorly aerated (-500 to -100 HU, in magenta) and nonaerated (-100 to +100 HU, dark red).

moderate and mild degree emphysema in the lower lobes (lung basis) and both lobes (hilum), respectively. These classification are in concordance with the classification based in lung function data.



(a) (b) (c)

Fig. 4. CT images of a healthy volunteer.On the left figure is shown the original CT images. In the center figure segmented images with the algorithm proposed by [16] and in the right figure result with the CMA to: hyperaerated (-1000 to -950 HU, light green), normally aerated (-950 to -500 HU, in blue), poorly aerated (-500 to -100 HU, in magenta) and nonaerated (-100 to +100 HU, dark red).

4 Conclusions

This paper proposes an algorithm that uses the pseudocolor segmentation techniques for the identification and quantification of pulmonary emphysema, in HRCT images of patients with COPD. The CMA is efficient in the identification and quantification, not only of regions with emphysema, but also of various ranges of lung densities with an accuracy of one pixel.

One advantage of the CMA is to present precise quantification indexes in percentage for all ranges of lung densities. Furthermore, the CMA highlights regions of the lungs that have subtle alterations in standard gray scale HRCT images that may be imperceptible to human vision.

Therefore, the CMA may have a wide application in improving the diagnoses and evaluation of emphysema extent, even in very early stages. Future works are needed in order to validate the CMA and to evaluate its incorporation in the clinical management of COPD patients.

Acknowledgments

The authors would like to thank to the Image and Pulmonology Departments of the Walter Cantídio University Hospital of the Federal University of Ceará.

References

1. da Costa e Silva, V.L., David, A.: Building blocks for tobacco control: a handbook, Geneva (2004)
2. Menezes, A.M.B., Jardim, J.R., Pérez-Padilla, R., Camelier, A., Rosa, F., Nascimento, O., Hallal, P.C., Team, P.: Prevalence of chronic obstructive pulmonary disease and associated factors: the platino study in são paulo, brazil. Cad. Saúde Pública 21, 1565–1573 (2005)

3. Celli, B., MacNee, W.: Standards for the diagnosis and treatment of patients with copd: a summary of the ats/ers position paper. European Respiratory Journal 23, 932–946 (2004)
4. Snider, G.L., Kleinerman, J., Thurlbeck, W.M., Bengali, Z.K.: The definition of emphysema: report of a national heart, lung and blood institute, division of lung diseases, workshop. Am. Rev. Respir. Dis., 182–185 (1985)
5. GOLD: Global initiative for chronic obstructive lung disease. global strategy for the diagnosis, management and prevention of chronic obstrutive pulmonary disease: Nhlbi/who workshop report. National Heart, Lung and Blood Institute, 100 (2001)
6. Stoel, B.C., Stolk, J.: Optimization and standardization of lung densitometry in the assessment of pulmonary emphysema. Investigative Radiology 39, 681–688 (2004)
7. Heard, B.E., Khatchatourov, V., Otto, H., Putov, N.V., Sobin, L.: The morphology of emphysema, chronic bronchitis, and bronchiectasis: Definition, nomenclature, and classification. Journal of Clinical Pathology, 882–892 (1979)
8. Coxson, H.O., Rogers, R.M., Whittall, K.P., D'Yachkova, Y., Paré, P.D., Sciurba, F.C., Hogg, J.C.: A quantification of the lung surface area in emphysema using computed tomography. Am. J. Respir. Crit. Care Med. 159, 851–856 (1999)
9. Madani, A., Keyzer, C., Gevenois, P.A.: Quantitative computed tomography assessment of lung structure and function in pulmonary emphysema. European Respiratory Journals 18, 720–730 (2001)
10. Müller, N.L., Coxson, H.: Chronic obstructive pulmonary disease.4: imaging the lungs in patients with chronic obstructive pulmonary disease. Thorax 57, 982–985 (2002)
11. Baldi, S., Miniati, M., Bellina, C.R., Battolla, L., Catapano, G., Begliomini, E., Giustini, D., Giuntini, C.: Relationship between extent of pulmonary emphysema by high-resolution computed tomography and lung elastic recoil in patients with chronic obstructive pulmonary disease. Am. J. Respir. Crit. Care Med. 164, 585–589 (2001)
12. Nixon, M.S., Aguado, A.S.: Feature Extraction and Image Processing, 1st edn. Newnes, Woburn (2002)
13. Gonzalez, R.C., Woods, R.: Digital Image Processing, 3rd edn. Pearson Prentice Hall, New Jersey (2008)
14. Gevenois, P.A., Vuyst, P.D., Maertelaer, V., Zanen, J., Jacobovitz, D., Cosio, M.G., Yernault, J.C.: Comparison of computed density and microscopic morphometry in pulmonary emphysema. American Journal of Respiratory and Critical Care Medicine 154, 187–192 (1996)
15. Vieira, S.R.R., Puybasset, L., Richecoeur, J., Lu, Q., Cluzel, P., Gusman, P.B., Coriat, P., Rouby, J.J.: A lung computed tomographic assessment of positive end-expiratory pressure-induced lung overdistension. Am. J. Respir. Crit. Care Med. 158, 1571–1577 (1998)
16. Felix, J., Cortez, P., Holanda, M., Costa, R.: Automatic segmentation and measurement of the lungs in healthy persons and in patients with chronic obstructive pulmonary disease in ct images. CLAIB 18, 370–373 (2007)

Computer-Guided Laparoscopic Training with Application of a Fuzzy Expert System

Andrzej Wytyczak-Partyka[1], Jan Nikodem, Ryszard Klempous, Jerzy Rozenblit[2], and Chuan Feng

[1] Wroclaw University of Technology,
Institute of Computer Engineering, Control and Robotics
27 Wybrzeze Wyspianskiego St., 50-370 Wroclaw, Poland
{andrzej.wytyczak-partyka,jan.nikodem,ryszard.klempous}@pwr.wroc.pl
[2] Electrical and Computer Engineering Department, The University of Arizona
Tucson, Arizona 85721, USA
{jr,fengc}@ece.arizona.edu

Abstract. Laparoscopic surgery is a widely accepted operating technique which continues to spread into different areas of medicine. Because of it's differences to open surgery (like limited perception) it demands a different training program than the traditional surgical training programs. Since it's introduction in 1980's several training curriculums for laparoscopic surgeons have been deployed and a set of skills that need to be mastered during the training has been defined. The training system proposed in this paper uses a knowledge base to guide the trainee trough the process of acquiring the necessary skills, based on the trainees measured performance in several areas. The system's guidance allows for better understanding of areas that need additional work and for faster acquisition of those, without the need for extra attention from the tutoring staff.

1 Introduction

Since it's raise in the early 1980's laparoscopic surgery has spread from cholecystectomy and appendectomy into many areas of surgery, including orthopaedic surgery, gynaecological surgery and others. The main reason behind it's growth are the significant benefits it brings into the healing process. Theese include: limited scarring, reduction in pain and recovery time, leading to a smaller risk of complications. Study conducted by Hansen et al. [1] shows that patients who have undergone laparoscopic appendectomy had five times fewer wound infections, two times shorter discharge time and fewer of them required narcotic analgesia.

On the other hand, there is a number of downsides, for instance the surgeon's perception both haptic and visual - is very limited, which extends the procedure time (in the open appendectomy case 40 as opposed to 63 minutes in laparoscopic appendectomy [1]) and the likelihood of human error. Also investment in expensive instruments and a very long training period are required.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 965–972, 2008.

There is a strong need, expressed in [2],[3],[4],[5], for computer assisted laparoscopic surgeon training. Reference [2] more thoroughly describes the motivation behind developing such a system.

The expert system described in this paper is designed to aid in the process of training laparoscopic surgeons within a computer-guided surgical training system and uses a fuzzy-logic reasoning as described in [6,7]. The simulator system used is based on the VAST trainer [8] and extends it as described in [2]. A structure-from-motion algorithm following [9] is used to recover the viewed scene's 3D geometry which then serves for defining no-fly zones within the scene.

The *no-fly zones* are an important concept used in this system. Those are the regions within the operating field where the appearance of an instrument is expected to inflict injury to the patient, i.e. nerves or vessels which shouldn't be irritated nor cut during the procedure. It is one of the goals of the presented system to teach the trainee to avoid such regions.

The simulator itself is a standard laparoscopic setting (camera and instrument) with a position sensor embedded at the tip of the instrument.

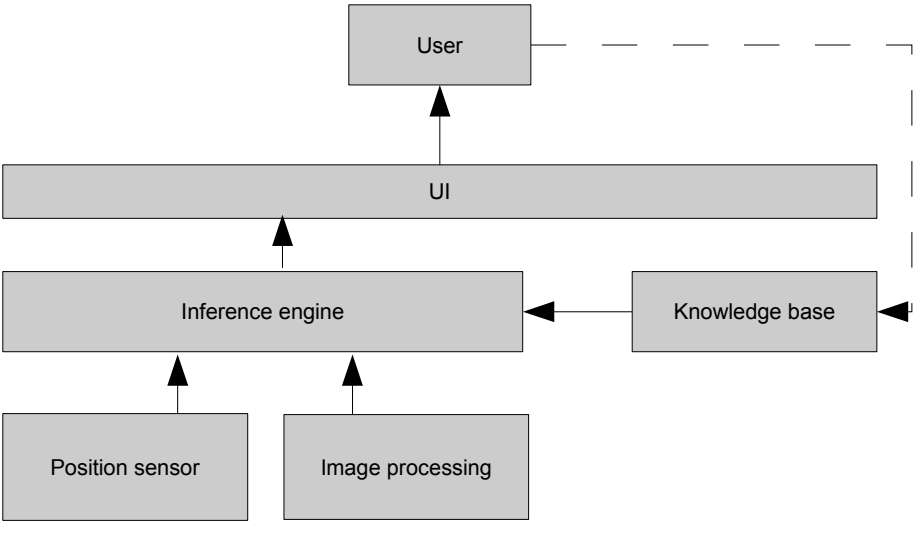


Fig. 1. Diagram of the proposed expert system

The system makes use of the information from a position sensor placed on the tip of the laparoscopic instrument and the 3D geometry of the scene to draw conclusions based on a set of rules from the system's knowledge base, Fig. 1. The image processing component of the system, described in more detail in Section 2, is used to build the 3D model for determining the no-fly zones. The 3D model is not used for visualisation purposes, but serves the inference engine for drawing conclusions about the users behaviour in relation to the no-fly zones. The system can then alert the user about aproaching a no-fly zone in a manner similar to

the parking sensor devices installed in modern cars, this and other means of interaction are described in Section 4.

2 Image Processing

The image processing component leads to obtaining a depth-map of the viewed scene from a series of images. Consider a following example of 2 input images of a training model as in Fig. 2. The image pair has been automatically selected from a set of views and next further processing has occured to obtain a depth-map representation of the 3D geometry of the scene, Fig. 3.

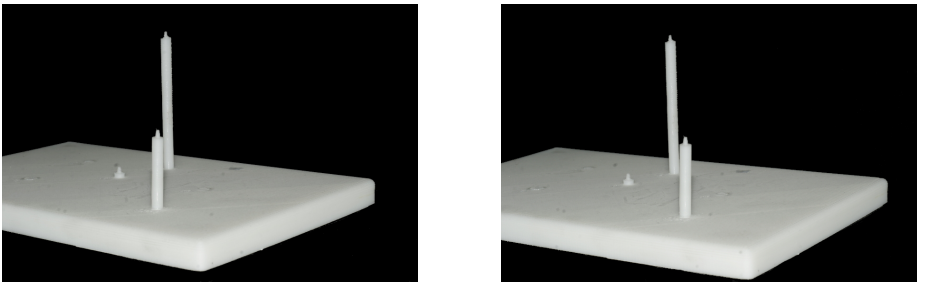


Fig. 2. 2 views of a training model used during exercises

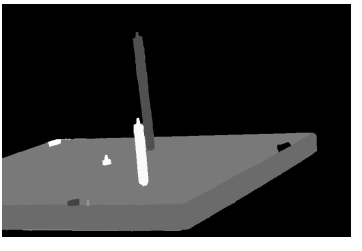


Fig. 3. Depth map image obtained with the structure-from-motion algorithm

The reconstructed approximation of the 3D relations of the scene are not aimed for visualisation, only for the step of defining no-fly zones. It is important to note, that the presented depth map has some flaws which are especially due to the lack of texture on the examined model.

3 Training

The training is based on simple tasks that trainees have to complete within a specified time. Several types of exercises have been proposed by others [10,11], the common goal is to practice dexterity, coordination and depth perception. Example exercies are :

– knot tying,
– cutting and suturing,
– picking up objects,
– touching different points on a model.

Each exercise is associated with a different physical model which is placed in the pelvi-trainer's box. The trainee's score is calculated with respect to :

– elapsed time
– length of the path of the instrument tip
– accuracy

and the score is calculated in a following manner :

$$S = \left(\frac{k_t}{t} + \frac{k_s}{s} + k_A \cdot A\right) \tag{1}$$

where k_i are the weights of i-th metric, t is the time metric, s is the path length metric, A is the accuracy metric (which is less or equal to 1).

As indicated in [2] another metric can be introduced, based on the 3D information obtained in the course of work of the structure-from-motion algorithm. This metric is used to determine the risk factor involved with the trainee's performance and it is related to breaches of the no-fly zones. The concept of this metric is that the score should be significantly decreased upon hitting a hazardous region (no-fly zone), defined for each exercise as a sphere with radius R with the center H_c in a certain point on the model

$$S = \left(\frac{k_t}{t} + \frac{k_s}{s} + k_A \cdot A + k_H \cdot H\right) \tag{2}$$

with

$$H = -\frac{2R_H}{|d(H_c, T)| + R_H} + 1, \tag{3}$$

and $d(H_c, T)$ being the distance between the center of the hazard sphere H_c and the instrument tip T. The H metric defined in such a way yields negative score for any position inside the no-fly zone, zero score for the tip on the sphere of the no-fly zone and positive score for anywhere outside the zone.

4 Inference Engine

The inference engine uses the dynamic data from the position sensor and image processing and the embedded knowledge from the knowledge base to conclude and inform the trainee about the areas where he or she underperforms during the exercise. The UI provides several means of communication with the user, i.e. the information about proximity to a no-fly zone can be presented as an auditory signal of a frequency that is dependent to the membership function defined in formula (5) creating an effect similar to the parking sensors embedded in car bumpers.

4.1 Movement Classification

In general the proposed engine deals with movements and their features, therefore a formalization of the term *movement* is needed. A movement m is defined by a pair of parameters (V, T), where V is the current speed vector and T the current position vector of the instrument's tip. Using such formalism movements can be divided into several classes, with respect to speed (F^+ set for fast movements and F^- for slow movements), path smoothnes (S^+ for smooth and S^- for rough ones), etc.

Arising from the metric introduced in Section 3 is the qualification of the movement as close to a no-fly zone, member of the H^+ set and far from a no-fly zone, member of H^- set. Further risky (belonging to the set R^+) and non-risky (the R^- set) movements can be defined, with regards to the distance from a no-fly zone, but also current speed of the movement.

We propose that the mentioned membership functions are not crisp, but fuzzy [12] - as it is hard to define i.e. a particular distance and speed at which a no-fly zone is safe from being breached. Let's consider the sets of risky and non-risky movements (R^+, R^-). The relation of a movement belonging to either R^+ or R^- shouldn't in fact be binary and can be expressed as a following fuzzy relation

$$m \in R^+ \iff (m \in H^+ \cap m \in F^+) \tag{4}$$

where H^+ and F^+ are fuzzy sets containing movements that are respectively, close to no-fly zones and fast. The membership functions for the H^+ and F^+ sets are as follows :

$$\mu_{H^+}(m) = \begin{cases} 0 & \text{for } d(H_c, T) > R_H + 1 \\ R_H - d(H_c, T) + 1 & \text{for } R_H < d(H_c, T) < R_H + 1 \\ 1 & \text{for } d(H_c, T) \leq R_H \end{cases} \tag{5}$$

$$\mu_{F^+}(m) = \begin{cases} 0 & \text{for } v < 0.1 \\ \frac{0.1 - V}{10} + 1 & \text{for } 0.1 < v < 10 \\ 1 & \text{for } v > 10 \end{cases} \tag{6}$$

where the parameter 1 in formula (5) and parameters 0.1, 10 in (6) are measured in cm and $\frac{cm}{s}$ respectively. In a similar manner we define the membership functions of the remainding movement classes.

4.2 Reasoning

Based on the classification of movements and the rules encoded in the knowledge base the system can evaluate the weaknesses of the trainee and provide simple advice on how to improve performance. The advice offered by the system is of the form

- your movements are too fast,
- your movements are too inprecise,
- your moving too close to a no-fly zone,

also the system can advise the user to go to a certain exercise which is designed to emphasise the particular skill the system considers to be weak. For instance the system can evaluate the movements as inprecise and can suggest the user to move to a training drill designed to practice the precision of movements.

5 Knowledge Base

The knowledge base of the expert system contains rules which are related to the movement classification, for example :

1. **If** the path between two points is not a straight line **then** the movement is not optimal.
2. **If** the instrument is in proximity to a no-fly zone **then** the movement is risky.
3. **If** the instrument moves very slow **then** the movement not optimal.
4. **If** the instrument moves very fast **then** the movement is risky.
5. **If** the instrument doesn't touch the specified points **then** the movement is inaccurate.
6. **If** the trainee's score is far lower than the average score **the** the overall performance was poor.

In addition a set of rules containing expert advice is available and presented to the user accordingly.

5.1 Feedback

The feedback mechanism of the presented system is such that the trainees' scores are recorded into the knowledge base. One of the consequences is that the trainee can be informed by the inference engine about how his performance stands among his peers. The recorded data also plays an important role in the development of the system as it indicates the weak points of the trainees and informs the developers about areas that need additional attention, i.e. during designing new exercises. After collecting a number of results from several trainees it is possible to see what are the common weaknesses and address them.

6 Preliminary Results

Brief analysis of the performance of the system's 3D recovery part shows that the accuracy of the Z-coordinate's reconstruction of the model pictured in Fig. 2 is far from perfect. After applying proper scale to the reconstructed model the maximum error of the depth estimation was in the range of $\pm 6mm$, which for the discussed model is 5% of it's total depth. The influence of the error on the system's performance can be partially minimized by enlarging the no-fly zones however it is clearly a significant flaw in the current version of the system.

The essence of the problem seems to lay in the last step of the 3D reconstruction process, which is stereo matching. During that step a pair of images, which

has been earlier matched and transformed, is examined to find pixel disparities for every pixel in an image. Several methods are known and a thorough analysis along with a testing method is given in [13]. Authors of the method selected for use in the presented system present promising results [14], even for poorly-textured models, however similar results couldn't be achieved for the model from Fig. 2.

7 Conclusions

The presented expert system based on fuzzy reasoning is a good solution for laparoscopic training programs based on computer-assisted simulators and application of the 3D reconstruction pipeline is a promising tool for more elaborate inference methods. A mathematical formalism for describing and classifying motion of the instrument's tip, which is strictly related to the trainee's skills, as well as a classification of movements has been given. The rules used by the inference engine to classify the movements and offer advice have been presented.

The image processing part of the system needs further attention, but already shows promising results. Especially models that are poorly textured produce 3D representations biased with a significant error, as it is the case with the model presented in this paper. Stereo matching algorithms that deal with such textureless objects are available and they will be examined for application in future versions of the proposed system.

References

1. Hansen, J.B.: Laparoscopic versus Open Appendectomy: Prospective Randomized Trial. World Journal of Surgery 20(1), 17–21 (1996)
2. Wytyczak-Partyka, A., Nikodem, J., Klempous, R., Rozenblit, J.: A novel interaction method for laparoscopic surgery training. In: Proceedings of the 8th Annual IEEE International Conference and Workshops on Human Systems Interactions, pp. 858–861 (2008)
3. Kneebone, R.: Simulation in surgical training: educational issues and practical implications. Medical Education 37(3), 267–277 (2003)
4. Rosser Jr., J.C., Rosser, L.E., Savalgi, R.S.: Objective Evaluation of a Laparoscopic Surgical Skill Program for Residents and Senior Surgeons. Archives of surgery 133(6), 657–661 (1998)
5. Rosser, J.C., Murayama, M., Gabriel, N.H.: Minimally invasive surgical training solutions for the twenty-first century. Surgical Clinics of North America 80(5), 1607–1624 (2000)
6. Ren, J., Sheridan, T.: Expert system training and control based on the fuzzy relation matrix. Technical report (1991), http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19910016327_1991016327.pdf
7. Kandel, A.: Fuzzy Expert Systems. CRC Press, Boca Raton (1992)
8. Feng, C., Rozenblit, J.W., Hamilton, A.J.: A Hybrid View in a Laparoscopic Surgery Training System. In: Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, pp. 339–348 (2007)

9. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual Modeling with a Hand-Held Camera. International Journal of Computer Vision 59(3), 207–232 (2004)
10. Derossis, A.M., Fried, G.M., Abrahamowicz, M., Sigman, H.H., Barkun, J.S., Meakins, J.L.: Development of a model for training and evaluation of laparoscopic skills. Am. J. Surg. 175(6), 482–487 (1998)
11. Fried, G.M., Feldman, L.S., Vassiliou, M.C., Fraser, S.A., Stanbridge, D., Ghitulescu, G., Andrew, C.G.: Proving the Value of Simulation in Laparoscopic Surgery. Annals of Surgery 240(3), 518–528 (2004)
12. Zadeh, L.A.: Fuzzy sets. World Scientific Series In Advances In Fuzzy Systems, 394–432 (1996)
13. Scharstein, D., Szeliski, R.: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. International Journal of Computer Vision 47(1), 7–42 (2002)
14. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: Proceedings of the 18th International Conference on Pattern Reconition, vol. 3, pp. 15–18 (2006)

Structural Sensitivity of Neural and Genetic Networks

Hedi Ben Amor[1], Jacques Demongeot[1], and Sylvain Sené[1,2]

[1] TIMC-IMAG, Faculty of medicine, University Joseph Fourier of Grenoble, 38706 La Tronche, France
[2] IXXI, Rhône-Alpes Complex Systems Institute, 5 rue du Vercors, 69007 Lyon, France

Abstract. This paper aims at giving new results on the structural sensitivity of biological networks represented by threshold Boolean networks and ruled by Hopfield-like evolution laws classically used in the context of neural and genetic networks. Indeed, the objective is to present how certain changes and/or perturbations in such networks can modify significantly their asymptotic behaviour. More precisely, this work has been focused on three different kinds of what we think to be relevant in the biological area of robustness (in both theoretical and applied frameworks): the boundary sensitivity (external fields, hormone flows, ...), the state sensitivity (axonal or somatic modulations, microRNAs actions, ...) and the updating sensitivity.

1 Introduction

Robustness studies in biological networks constitute now a real challenge on both views *i*) of the evolution [1,2]: what kind of structures insensitive to environmental perturbations have been selected? and *ii*) of the function [3,4]: which type of critical part of a regulatory network is concerned by its function? The first point concerns the speciation with inheritance of interaction architectures having already shown their robustness and the second concerns the differentiation, *i.e.*, the ability to create new functionalities, corresponding to new dynamical regimes of the network fulfilling new functions by optimising new local cost functionals.

An evolutionary view on inheritance of regulatory networks leads to build phylogenetic trees of networks by using adequate distances between their interaction graphs in order to calculate barycenters of clusters and extract putative ancestors [5]. A way to validate these trees could be to follow the increasing robustness of their leaves and a good characterization of this robustness could be a score incorporating three kinds of sensitivity: boundary sensitivity (to external chemicophysical fields like substrate input or to external controllers like hormonal morphogenetic or growth factors), state sensitivity (to endogeneous action like neuronal axonal modulations or microRNAs inhibition) and updating sensitivity (to changes in dynamical behaviours induced by the mode of updating, parallel,

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 973–986, 2008.

sequential, etc.). Each sensitivity can be quantified by the mean probability to change of stability basin under the corresponding perturbation [6,7].

So, in this paper, after having presented the major useful definitions in Section 2, we give in Section 3 new results about the influence of boundary conditions in stochastic Hopfield-like networks. More precisely, we focus on the differences between the impact of extremal and "quincuncial" boundaries on theoretical networks, *i.e.*, lattices on $\mathbb{Z}^2$. In Section 4, we consider real regulatory networks, one devoted to the control of hair morphogenesis in mice and the second to the cell cycle control in superior eukaryotes, and we show in which way the addition of critical nodes in these networks can perturb their attractors.

2 Preliminary Definitions

We consider in the following Hopfield-like networks [8]. More precisely, we denote by R a connected biological network of N nodes having two possible activity states such that $\sigma_i(t) = 0$ (resp. 1) if node i is inactive (resp. active) at time t. We call $\sigma(t) = (\sigma_i(t))_{i \in R} \in \Omega = \{0,1\}^N$ the configuration of the network at time t, where Ω is the set of all possible configurations of R. Such networks can be represented by oriented graphs in which vertices correspond to the nodes and the directed edges to the interactions between these nodes. Each interaction is characterised by a potential which gives the weight of the action that the source node gets on the destination node. Being given such an arbitrary network R, we associate to it an interaction matrix $W_{N \times N}$ whose coefficient w_{ij} corresponds to the interaction potential that node j exerts on node i. More precisely, the coefficient w_{ij} can be positive or negative depending on the fact that node j tends to respectively activate or inhibit node i and is null if j has no influence on i. In these networks, the centre is defined by the set of nodes whose eccentricity (*i.e.*, maximal length of the shortest paths to reach from a node all the others) is minimal and the boundary by the set of Gardens of Eden (*i.e.*, nodes whose inner degree is null, the degree of a node being the number of edges attached to it) whose distance to the center is greater than the minimal distance from the center to all its accessible pendant edges (*i.e.*, nodes whose outer degree is null). Note that if there is not such pendant node, the boundary of the network is the set of all Gardens of Eden. In the sequel of this article, we will focus on two different kinds of Hopfield-like neural networks whose evolution is ruled either by a deterministic law or by a stochastic law. Let us now present these two evolution laws.

Let $\mathcal{N}_i$ be the neighbourhood of node i, *i.e.*, the set of nodes j's having an influence on i. More formally, $j \in \mathcal{N}_i \Leftrightarrow w_{ij} \neq 0$. Let us also denote by $H(\sigma_i(t))$ the interaction potential of node i at time t. $H(\sigma_i(t))$ is then defined by:

$$H(\sigma_i(t)) = \sum_{j \in \mathcal{N}_i} w_{ij} \cdot \sigma_j(t) - \theta_i$$

where θ_i is the activation threshold of node i, *i.e.*, the quantity of interaction potential that has to be overtaken in order to node i becomes (or stays) active.

From this interaction potential, we define respectively as follows the deterministic (rule 1) and the stochastic (rule 2) Hopfield-like evolution rules:

$$\sigma_i(t+1) = \mathcal{H}(H(\sigma_i(t))) \tag{1}$$

where $\mathcal{H}$ is the Heaviside (or sign-step) function such that $\mathcal{H}(x) = 0$, if $x \leq 0$, and 1 otherwise, and

$$P(\sigma_i(t+1) = \alpha \mid \sigma_j(t), j \in \mathcal{N}_i) = \frac{e^{\alpha \cdot H(\sigma_i(t))/T}}{1 + e^{H(\sigma_i(t))/T}} \tag{2}$$

where T is the temperature of the system.

Let us remark that the temperature allows to render the system more or less probabilistic. Indeed, when the temperature tends to 0, the stochastic rule is equivalent to the deterministic one and, when the temperature tends to infinity, the probability for the state of a node to be equal to 1 is $\frac{1}{2}$.

3 Extremal and "Quincuncial" Boundaries in 2D Theoretical Networks

The comprehension of the behaviour of real biological regulatory networks is a difficult problem that researchers, from many different disciplines, are now more and more to study. Some works have highlighted the particular topology of biological networks such as neural and genetic regulation networks [9,10]. Nevertheless, their behaviour can be sometimes better understood by studying more theoretical networks such as the ones of cellular automata [11]. However, most researches have been focused on the comprehension of emergent phenomena in infinite and periodic (toric) cellular automata [12,13], which could be criticised from a biological point of view. Indeed, living systems do not develop infinitely and each of their components, *e.g.*, a cell, is confined in a large but finite space. That is why, in this part, we decide to focus on finite systems, with fixed boundaries surrounding them. More precisely, in [14], the authors present simulation results highlighting that the influence of fixed boundary conditions can lead to the emergence of phase transitions in the asymptotic behaviour of stochastic Hopfield-like networks and compare the obtained results with those known about magnetic networks. In this section, with the same method, we decide to go further and show that the nature of the deployed fixed boundary conditions can affect quantitatively these phase transitions.

3.1 Definitions and Simulation Protocol

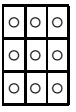
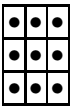
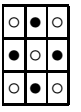
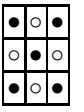
The networks on which we focus here are represented by two-dimensional square lattices whose elements are located at vertices of $\mathbb{Z}^2$ and are governed by the stochastic evolution law. More formally, if we denote such a network composed of N nodes by R, it is defined by: $R = ([-k, \ldots, k] \times [-k, \ldots, k]) \cap \mathbb{Z}^2$ and $N = (2 \cdot k + 1)^2$. Furthermore, the neighbourhood of a node i belonging to

R is the set of five nodes composed by i itself and its four nearest neighbours j_1, j_2, j_3, and j_4 such that $\forall k \in \{1, ..., 4\}, d(i, j_k) = 1$. Likewise in [14], we define the boundary of such a network by the set of nodes of outer degree less or equal than 4. In the same way, we define the centre c of the network by the node being at equal distance of the four corners of the lattice. Let us remark that these definitions of boundary and centre can be generalised to arbitrary oriented graphs by using the definition of graph eccentricity as in [6,7] and that they are also in respect with the ones given in Section 2. As it has been introduced above, the objective is to compare the effects on the asymptotic behaviour of such networks induced by fixed boundary conditions of different nature. In order to have more chance to obtain significant results on this subject, we study the two farthest kinds of boundary conditions in the spectrum of possibles, *i.e.*, the extremal and the quincuncial ones. By stretching out the boundary conditions by beginning from the upper left corner of the lattice and going clockwise, we can represent them by vectors of size $4 \cdot \sqrt{N} - 4$. The extremal ones (denoted by B_e) are then distinguished in two types: $B_e^{(0)} = \{0, 0, \ldots, 0\}$ and $B_e^{(1)} = \{1, 1, \ldots, 1\}$ as well as the quincuncial ones (denoted by B_q): $B_q^{(01)} = \{0, 1, 0, 1, \ldots, 0, 1\}$ and $B_q^{(10)} = \{1, 0, 1, 0, \ldots, 1, 0\}$. We define the activity of a node i by the number of time iterations at which i is active. The measure of the influence of fixed boundary conditions is based on the computation of the asymptotic activity at the centre of the network. Let us denote by T_t the transient time during which the system evolves to reach its asymptotic behaviour and by T_s the sampling time during which the activity of the centre is computed. Thus, the central activity of a network is defined by: $S = \sum_{t \in T_s} \sigma_c(t)$ and corresponds consequently to the number of sampling time iterations during which the center c has been observed active. Depending on the chosen fixed boundary conditions, we can easily obtain $S^{(0)}$, $S^{(1)}$, $S^{(01)}$ and $S^{(10)}$. The measures (computed thanks to Monte-Carlo simulations) are then respectively for extremal and quincuncial boundaries: $S_e = \frac{|S^{(0)} - S^{(1)}|}{|T_s|}$ and $S_q = \frac{|S^{(01)} - S^{(10)}|}{|T_s|}$. It has been shown in [14] that, in certain domains of the parameters $u_0 = \frac{w_{ii}}{T}$ and $u_1 = \frac{w_{ij}}{T}$ and for networks sizes of different order of magnitude, the influence of fixed boundary conditions is significant and leads to the emergence of phase transitions. In this work, we focus on homogeneous (isotropic and translation invariant) attractive ($u_1 > 0$) and repulsive ($u_1 < 0$) networks. Since the existence of phase transitions has been highlighted in [14], we simulate the asymptotic behaviour of small 2D networks of size 11×11 and execute a Monte-Carlo algorithm on 500 different initial configurations of density 0.5, *i.e.*, the probability for a node to be activated at time $t = 0$ is equal to $\frac{1}{2}$.

3.2 Results

Let us now present some new results concerning the correlations between the influence and the nature of fixed boundary conditions. We emphasise the higher influence of extremal boundaries in attractive Hopfield-like networks before highlighting the significant impact of quincuncial boundaries in repulsive networks.

To do that, we focus on the measures S_e and S_q defined above and on the specific probabilities to observe (in the phase transitions domains of parameters u_0 and u_1) the following extremal and quincuncial patterns over sampling time on the central square of size 3×3:

$p^0 =$  $p^1 =$  $p^{01} =$  $p^{10} =$ 

In particular, we emphasise that the computed measures S_e and S_q are correlated to these probabilities that give a statistical explanation of the quantitative differences observed. We define by P_0^0 (resp. P_1^0, P_{01}^0, P_{10}^0) the probability to observe the pattern p^0 when boundaries states are fixed to 0 (resp. 1, the quincunx 01 and the quincunx 10). In the same way, we define by P_x^1 (resp. P_x^{01}, P_x^{10}) the probability to observe p^1 (resp. p^{01}, p^{10}) where x can be 1, 01 or 10. The probability measure, depending on the probabilities to observe specific patterns, relevant to establish the correlation introduced above, is the minimal proportion $\mathcal{P}$ of sampling time iterations on which different patterns (specific or not) can be observed. Indeed, observing different patterns is a necessary condition to observe values variations of the measures $S^{(0)}$ and $S^{(1)}$ or $S^{(01)}$ and $S^{(10)}$. In the logic terminology, the event $\mathcal{E}$ of probability $\mathcal{P}$ is defined in the extremal (resp. quincuncial) boundaries case by $\mathcal{E}_e = \neg((E_0^0 \wedge E_1^0) \vee (E_0^1 \wedge E_1^1) \vee (E_0^{01} \wedge E_1^{01}) \vee (E_0^{10} \wedge E_1^{10}))$ (resp. $\mathcal{E}_q = \neg((E_{01}^0 \wedge E_{10}^0) \vee (E_{01}^1 \wedge E_{10}^1) \vee (E_{01}^{01} \wedge E_{10}^{01}) \vee (E_{01}^{10} \wedge E_{10}^{10})))$ where E_x^p denotes the event of probability P_x^p to observe the pattern p with the boundary x. Since we want to obtain probabilities lower bounds, the probabilities $\mathcal{P}_e$ and $\mathcal{P}_q$ are defined by complement to 1 of the maximal probabilities to have a specific pattern and we have the following lower bounds:

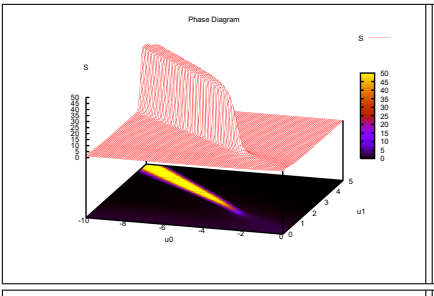
$$\mathcal{P}_e \geq 1 - [\sum_{x=0,\ y=1} \min(P_x^0, P_y^0) + \ldots + \min(P_x^{10}, P_y^{10})]$$

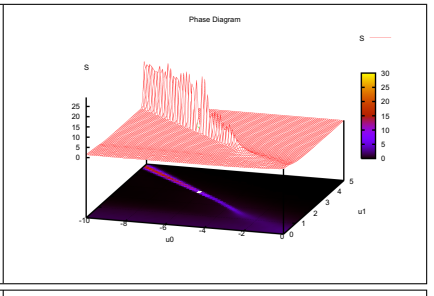
$$\mathcal{P}_q \geq 1 - [\sum_{x=01,\ y=10} \min(P_x^0, P_y^0) + \ldots + \min(P_x^{10}, P_y^{10})]$$

Let us notice that the standard deviations of the $\mathcal{P}$'s are the sum of the four maximal standard deviations of the couples (P_x^p, P_y^p) where $p = 0, 1, 01, 10$ and that for $(x, y) = (0, 1)$ or $(01, 10)$.

Let us first focus on attractive networks. The figures presented in Table 1 show that the influences exerted by extremal and quincuncial fixed boundary conditions are qualitatively very close. Indeed, the two kinds of boundary conditions bring to the emergence on the straight line $u_0 + 2 \cdot u_1 = 0$ of a phase transition that has been studied in [14]. However, the emergence of phase transition is quantitatively significantly more important with extremal than with quincuncial boundary conditions, which can be explained by the probabilities introduced above and presented in the second part of Table 1. Before studying these probabilities, let us give some intuition about the behaviour of such networks on the transition line with extremal and quincuncial boundaries. First, in

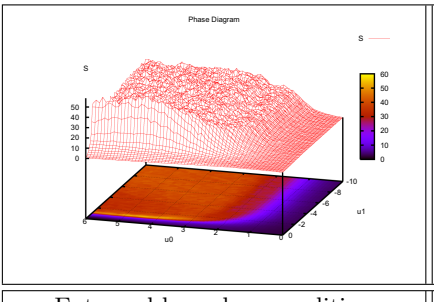
Table 1. Simulation results obtained with attractive Hopfield-like networks presenting the values of the two measures S_e and S_q and, below, the probabilities (with their 95% confidence intervals) on the phase transition line ($u_0 + 2 \cdot u_1 = 0$) to observe specific patterns on the central square

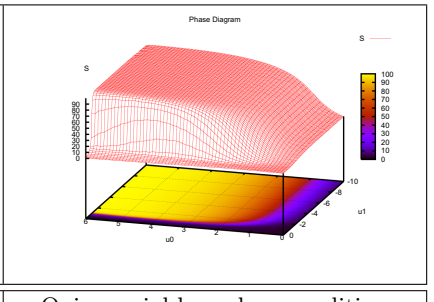




Extremal boundary conditions				Quincuncial boundary conditions			
P_0^0	$0.00 \pm 3 \cdot 10^{-5}$	P_1^0	$0.00 \pm 2 \cdot 10^{-5}$	P_{01}^0	$0.00 \pm 3 \cdot 10^{-5}$	P_{10}^0	$0.00 \pm 2 \cdot 10^{-5}$
P_0^1	$0.02 \pm 5 \cdot 10^{-4}$	P_1^1	$0.63 \pm 2 \cdot 10^{-3}$	P_{01}^1	$0.57 \pm 2 \cdot 10^{-3}$	P_{10}^1	$0.55 \pm 2 \cdot 10^{-3}$
P_0^{01}	$0.31 \pm 2 \cdot 10^{-3}$	P_1^{01}	$0.00 \pm 1 \cdot 10^{-4}$	P_{01}^{01}	$0.02 \pm 6 \cdot 10^{-4}$	P_{10}^{01}	$0.03 \pm 7 \cdot 10^{-4}$
P_0^{10}	$0.31 \pm 2 \cdot 10^{-3}$	P_1^{10}	$0.00 \pm 2 \cdot 10^{-4}$	P_{01}^{10}	$0.03 \pm 6 \cdot 10^{-4}$	P_{10}^{10}	$0.03 \pm 7 \cdot 10^{-4}$

Table 2. Simulation results obtained with repulsive Hopfield-like networks. The reading of these results is the same than in Table 1.





Extremal boundary conditions				Quincuncial boundary conditions			
P_0^0	$0.00 \pm 5 \cdot 10^{-6}$	P_1^0	$0.00 \pm 6 \cdot 10^{-6}$	P_{01}^0	$0.00 \pm 8 \cdot 10^{-6}$	P_{10}^0	$0.00 \pm 3 \cdot 10^{-6}$
P_0^1	0.00 ± 0.00	P_1^1	0.00 ± 0.00	P_{01}^1	0.00 ± 0.00	P_{10}^1	0.00 ± 0.00
P_0^{01}	$0.24 \pm 2 \cdot 10^{-4}$	P_1^{01}	$0.23 \pm 2 \cdot 10^{-4}$	P_{01}^{01}	$0.80 \pm 2 \cdot 10^{-4}$	P_{10}^{01}	$0.00 \pm 7 \cdot 10^{-7}$
P_0^{10}	$0.49 \pm 2 \cdot 10^{-4}$	P_1^{10}	$0.53 \pm 2 \cdot 10^{-4}$	P_{01}^{10}	0.00 ± 0.00	P_{10}^{10}	$0.76 \pm 2 \cdot 10^{-4}$

the case of extremal boundaries, we can think intuitively that the active boundary nodes tend to activate their neighbours. Once active, these neighbours tend to activate their own neighbours and, by induction, tend to propagate the active state until the centre of the network. Thus, an active boundary tends to create an important proportion of patterns p^1 on the central square. Furthermore, the intuition in the case of an inactive boundary is that, since the latter has no influence and since we are on the transition line, no nodes are fixed to an

active state and the activating interaction potentials w_{ij}'s in the networks are counter-balanced by the inhibiting interaction potentials w_{ii}'s (autoinhibitions). Consequently, there are only few chances that two neighbour nodes are active together. Thus, the most probable patterns are the quincuncial ones p^{01} and p^{10}. Eventually, quincuncial boundaries should create an important proportion of patterns p^1 since the fixed active nodes tend to activate their neighbourhood in particular for important values of u_1. Let us remark that all these intuitions are confirmed by the computed probabilities (see Table 1).

To conclude on attractive networks, let us note that: (i) in the quincuncial boundaries case, the minimal probability $\mathcal{P}_q$ for which significant values of the measure S_q can be observed is $\mathcal{P}_q = 1 - (0.55 + 0.02 + 0.03) = 0.40$; (ii) in the case of extremal boundaries, $\mathcal{P}_e$ is equal to $1 - (0.02) = 0.98$. Consequently, in attractive networks, on the phase transition line, an event at which S_e is incremented has at least $\frac{0.98}{0.40} \approx 2.45$ times more chances to occur than an event at which S_q is incremented.

In the case of repulsive networks, the same kind of phenomenon appears. Indeed, the "parabolic" phase transitions (see [14]) induced by the influence of boundary conditions do not depend qualitatively on the nature of boundaries: with extremal and quincuncial boundaries, their parametric domain of emergence stay similar. However, here also, significant quantitative differences appear and, contrary to the attractive case, quincuncial boundaries appear to have more impact (see Table 2). Before explaining this phenomenon and because the behaviour of repulsive networks are less intuitive, let us show why quincuncial boundaries tend to lead their own pattern until the central square of the network.

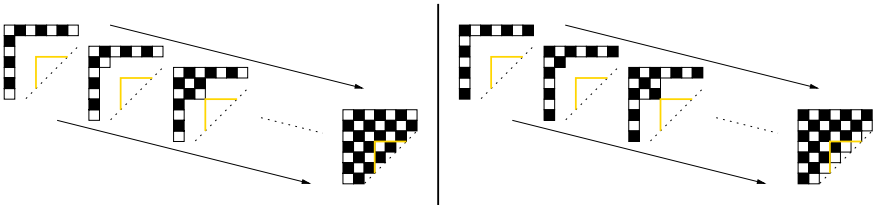


Fig. 1. Construction of a repulsive network with quincuncial boundaries

Proposition 1. *In a repulsive Hopfield-like network represented by a square lattice of odd side size in $\mathbb{Z}^2$, quincuncial boundaries reproduce in majority their own pattern at the centre of the network.*

Proof. The proof is done by induction on a square lattice of odd side size n and by considering the assumption that the quincuncial boundary is $B_q^{(01)}$. The aim is to show how to build the most probable configuration being only given the boundary. Let us consider at first time the square K_1^q of size 2×2 in the upper left corner of the lattice $R \cup \partial R$, where ∂R is the boundary of R. As we know the states of boundary nodes, we have to evaluate the state of the node n_1 located

at the bottom right corner of K_1^q. The known nearest neighbours of this node are active (by hypothesis on the nature of the boundary). Since the network is repulsive, these two nodes tend to inhibit n_1. Let us now consider the square K_2^q of size 3×3 in the upper left corner of the lattice for which we know the state of six nodes. Let us focus on the middle right (n_2) and bottom centre (n_3) nodes of K_2^q. Their known nearest neighbours are inactive. Thus, because of the counter-balancing property on the phase transition parametric domain, n_2 and n_3 tend to be activated by their own activating retroaction potential and, consequently, as it was the case for K_1^q, they tend to inhibit their nearest neighbour n_4 located at the bottom right corner of K_2^q. By induction, if we build the network of side size n by constructing step by step upper left squares of side size bigger and bigger, one remark that all the inner squares of odd side size whose centre is the centre of the network contain exactly the pattern of the boundary. It is easy to show that the result stays true if we change boundary $B_q^{(01)}$ by $B_q^{(10)}$ (see Figure 1). Thus, we have the expected result. □

The probabilities given in Table 2 show that the most probable central pattern with extremal boundaries is p^{10}. Let us now explain these probabilities and show why the extremal boundaries tend to produce the pattern p^{10} on the central square of repulsive networks.

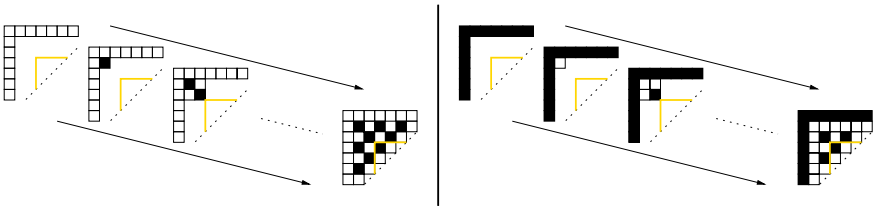


Fig. 2. Construction of a repulsive network with extremal boundaries for important inhibiting interaction potentials

Proposition 2. *In a repulsive Hopfield-like network represented by a square lattice of odd side size in $\mathbb{Z}^2$, extremal boundaries reproduce in majority the quincuncial pattern p^{10} until the centre of the network if the inhibiting interaction potentials are greater than the activating retroaction potentials.*

Proof. Here also, the proof is performed by induction on a square lattice of odd side size n and by considering extremal boundaries $B_e^{(1)}$. Let us perform the proof by building the most probable configuration being only given the boundary. Let us consider the square K_1^e of size 2×2 in the upper left corner of the lattice. The node n_1 at the bottom right corner of K_1^e is surrounded by two active nodes which tend to inactive n_1. Let us now consider the square K_2^e of size 3×3 in the upper left corner of the lattice. Both the middle right and bottom centre nodes n_2 and n_3 have one active neighbour. Since we focus on repulsive networks and have done the hypothesis that the inhibiting interaction potentials are sufficiently

important, they both tend to be inactivated. Thanks to this hypothesis, it easy to remark that all the neighbours of the boundary nodes are inactivated by the latter. Let us continue with the construction of K_2^e and consider the node n_4 located at the bottom right corner of K_2^e. At this step, n_4 is only surrounded by two inactive nodes. Thus, as no node exerts an inhibiting potential on it, n_4 has good chance to activate itself thanks to its retroaction potential and will tend to inhibit its other neighbours at next step. By induction, if we build the network of side size n by constructing step by step upper left squares of side size bigger and bigger, one remark that, except for the central square of side size $n-2$, all the other central squares of odd side size contain exactly the pattern p^{10}. Let us remark that the proof for the extremal boundary $B_e^{(0)}$ is captured here (see Figure 2). Consequently, we have the expected result. □

This proposition is based on the hypothesis that the inhibiting interaction potentials are greater than the activating retroaction potentials. However, the probabilities obtained by simulations have been computed for the whole phase transition domain in which the values of the parameter u_0 are greater than those of u_1 approximatively in 50 percent of the cases. Consequently, it seems that this proposition stay true even without the hypothesis. Besides, we have checked this assumption by computing the probabilities to observe the specific patterns p^0, p^1, p^{01} and p^{10} on the domain of phase transition delimited by $4 \leq u_0 \leq 5$ and $-2.5 \leq u_1 \leq -2$. The obtained probabilities are:

P_0^0	P_0^1	P_0^{01}	P_0^{10}
0.00 ± 0.00	0.00 ± 0.00	$0.37 \pm 2 \cdot 10^{-3}$	$0.52 \pm 2 \cdot 10^{-3}$
P_1^0	P_1^1	P_1^{01}	P_1^{10}
0.00 ± 0.00	0.00 ± 0.00	$0.28 \pm 2 \cdot 10^{-3}$	$0.59 \pm 2 \cdot 10^{-3}$

and show that the hypothesis is not necessary for Proposition 2 to be valid.

To conclude on repulsive networks, let us note that: (i) in the extremal boundaries case, the minimal probability $\mathcal{P}_e$ for which significant values of the measure S_e can be observed is $\mathcal{P}_e = 1-(0.23+0.49) = 0.28$; (ii) in the case of quincuncial boundaries, $\mathcal{P}_q$ is equal to 1. Consequently, in repulsive networks, on the phase transition line, an event at which S_q is incremented has at least $\frac{1}{0.28} \approx 3.57$ times more chances to occur than an event at which S_e is incremented.

4 Architecture of Real Genetic Regulation Networks

In this section, we will show that the previous results on the dependence of core states on the boundary states of the interaction network are also observed in real regulatory networks. We will use for that two simple real genetic networks, the first related to the control of the cell cycle in superior eukaryotes and the second to the regulation of hair morphogenesis in mice. Since we have no particular indication about the interaction weights, we fix their absolute values to 1 and the thresholds to 0 and use the deterministic Hopfield-like evolution law.

4.1 Cell Cycle

The genetic network controlling the cell cycle in superior eukariotes cells [15,16] possesses a core made of the gene Rbp-E2F (whose eccentricity is equal to 2) whose strongly connected component is down-regulated in human by one micro-RNA (miRNA 159 acting on the transcription factor E2F). The frontier of its interaction graph contains the two elements Cdk2 and miRNA 159. For this study, we have focused on the impact of two nodes: the boundary node miRNA 159 because of its direct interaction on the central strongly connected component and the protein p27 since it has the interesting characteristic to be the node of maximal eccentricity (= 7) in the interaction graph (see Figure 3). Note that the interaction graph contains only one connected component having at least one (here two) positive circuit of interactions (a circuit is positive if its number of inhibiting edges is even). Hence, from [17,18,19,20,21,22,23], we can expect only $2^1 = 2$ fixed configurations for the network dynamics and an upper bound for this number of 2^2. On Table 3, we see that, if the state of p27 and miRNA

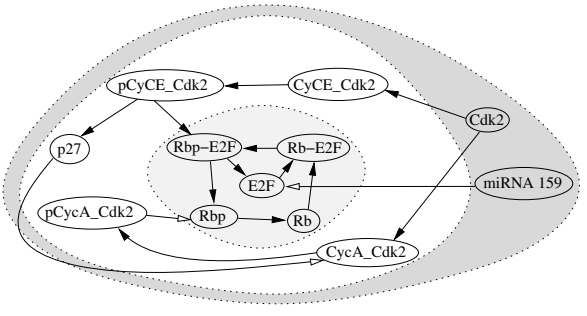


Fig. 3. Genetic regulation network controlling the cell cycle in eukariotes cells (the strongly connected component containing the central node Rbp-E2F is in a light grey space and the boundary node is in a dark grey space)

Table 3. Attractors and attraction basins relative sizes (ABRS) of the cell cycle network dynamics for the sequential and parallel updating iteration modes. The nodes are ordered as follows: p27, Cdk2, pCyCE_Cdk2, CyCE_Cdk2, miRNA 159, pCycA_Cdk2, CycA_Cdk2, Rbp-E2F, Rb-E2F, E2F, Rbp and Rb. The attractor average distance (AD) is the average number of transitions needed from an initial configuration to reach the attractor ; this average distance is correlated with the ABRS.

	Sequential updating		Parallel updating		
Nature	Attractor	ABRS	Attractor	ABRS	AD
Fixed point 1	000000000000	6.25%	000000000000	0.5%	1.45
Fixed point 2	000000011111	56.25%	000000011111	99.5%	5.40
Limit cycle 1	000000001000 000000010111	37.5%	None		

159 are not fixed to particular values, then this number is in reality 2, plus one (resp. zero) limit cycle in the case of sequential (resp. parallel) updating mode. The conjecture that the number of fixed configurations is equal to 2^m, where m is the number of connex components having at least one positive circuit is here available, as well as the upper bound. The asymptotic result [23] claiming that the number of fixed configurations is of the order of magnitude of the square root of the network size (here $\sqrt{12}$) is also verified. If the boundary state of p27 is fixed to 1, then we keep in the parallel case the two fixed configurations. If the state of miRNA 159 is fixed to 1, due to its constant inhibition, we observe 6 attractors in the parallel case:

Nature	Attractor	ABRS
Fixed point 1	000010000000	0.12%
Fixed point 2	000010011011	0.54%
Limit cycle 1	000010001010 000010010001	10.6%
Limit cycle 2	000010010011 000010001011 000010011001 000010011010	26.6%
Limit cycle 3	000010010010 000010000011 000010001001 000010011000	32.4%
Limit cycle 4	000010000010 000010000001 000010001000 000010010000	23.7%

If we fix both p27 and miRNA 159 to 1, we keep the previous attractors.

4.2 Mice Hair

The genetic network controlling the hair morphogenesis in mice [24] has a core containing 2 genes, β-catenin and Cyclin D1 (whose eccentricity is equal to 5, see Figure 4) with a motif called incoherent feed-forward [25]. Its boundary is composed by the five following nodes: Smad3, miRNA 141, EphA3, SrC and Zfhx3. In the following, we are going to focus on the influence induced by the two boundary nodes of maximal eccentricity ($= 9$) which have two different actions on the network. Indeed, the strongly connected component containing the core is down-regulated in human by the micro-RNA 141 [26], acting on the protein p53, and up-regulated by the protein phosphatase EphA3. The interaction graph contains only one connected component having at least one (here five) positive circuits of interactions, as in the previous example of the cell cycle, we expect only $2^1 = 2$ fixed configurations for the network dynamics and an upper bound of 2^5. On Table 4, we see that, if the boundary state of miRNA 141 is not

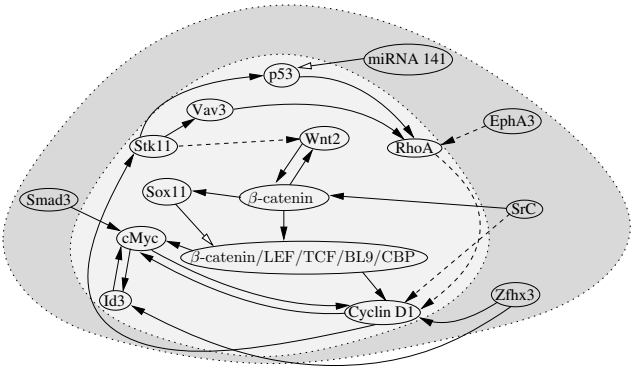


Fig. 4. Genetic regulation network controlling the hair morphogenesis in mice (the strongly connected component containing the central nodes is in a light grey space and the boundary nodes are in a dark grey space)

Table 4. Attractors and attraction basins relative sizes (ABRS) of the dynamics of the network modelling the hair morphogenesis of mice for the sequential and parallel updating iteration modes. The nodes are ordered as follows: miRNA 141, EphA3, p53, Vav3, Stk11, Wnt2, RhoA, Smad3, SrC, Id3, Cyclin D1, Zfhx3, Sox11, β-catenin, cMyc and β-catenin/LEF/TCF/BL9/CBP.

	Sequential updating		Parallel updating		
Nature	Attractor	ABRS	Attractor	ABRS	AD
Fixed point 1	0000000000000000	1.56%	0000000000000000	$\approx 0.00\%$	0.75
Fixed point 2	0011111001101110	96.88%	0011111001101110	99.66%	4.14
Fixed point 3	0000010000001100	1.56%	0000010000001100	$\approx 0.00\%$	0.5
Limit cycle 1	None	–	0011010001101100 0000111000001110	0.34%	2.45

fixed to a specific value, then this number is in reality 3, plus one limit cycle in the case of parallel updating mode, which disappears by introducing delays in the graph for the indirect interactions (arrows in dashed lines on Figure 4) for which we introduce each time an intermediary node (*i.e.* one node between Stk11 and Wnt2, one between RhoA and Cyclin D1 and one between SrC and Cyclin D1). The conjecture introduced above is false because the number m of connex components having at least one positive circuit is not sufficient, but the upper bound built from the number p of positive circuits stays correct, and we will conjecture for the future that the number of attractors for a Hopfield-like network relies between the two limits 2^m and 2^p. The asymptotic (in the number of genes) result claiming that the number of fixed points is of the order of magnitude of the square root of the network size (here $\sqrt{16}$) is also verified. If the boundary state of miRNA 141 is equal to 1, then we keep in the parallel case the same attractors by chanching the state of miRNA 141 from 0 to 1

(we only have the state of p53 constantly equal to 0 due to the inhibition), conserved (except the limit cycle) by adding delays. On the contrary, by changing the state of EphA3 from 0 to 1, we have only one fixed configuration in the parallel case: 0111111001101110, conserved by adding delays.

5 Conclusion

We have studied, in this paper, the influence of boundaries on Hopfield-like regulatory networks: this influence is present in case of quincuncial boundaries for neural networks especially when the interactions are repulsive. The sensitivity in real genetic networks appears also dominant in the case of inhibitory actions exerted by micro-RNAs. A more systematic study could be processed in the future in order to confirm this dominant influence of negative interactions, for which the Hopfield-like regulatory networks seem to be less robust than for the positive ones (activations).

References

1. Maslov, S., Sneppen, K., Eriksen, K.A., Yan, K.K.: Upstream Plasticity and Downstream Robustness in Evolution of Molecular Networks. BMC Evolutionary Biology 4(9) (2004)
2. Handorf, T., Ebenhöh, O., Heinrich, R.: Expanding Metabolic Networks: Scopes of Compounds, Robustness, and Evolution. Journal of Molecular Evolution 61(4), 498–512 (2005)
3. Kitano, H.: Towards a Theory of Biological Robustness. Molecular Systems Biology 3 (2007)
4. Kitano, H.: Biological Robustness in Complex Host-Pathogen Systems. Progress in Drug Research 64(239), 241–263 (2007)
5. Ye, P., Peyser, B.D., Spencer, F.A., Bader, J.S.: Commensurate Distances and Similar Motifs in Genetic Congruence and Protein Interaction Networks in Yeast. BMC Bioinformatics 6, 270 (2005)
6. Demongeot, J., Morvan, M., Sené, S.: Impact of Fixed Boundary Conditions on the Basins of Attraction in the Flower's Morphogenesis of Arabidopsis Thaliana. In: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications Workshops, pp. 782–789. IEEE Computer Society Press, Los Alamitos (2008)
7. Demongeot, J., Morvan, M., Sené, S.: Robustness of Dynamical Systems Attraction Basins Against State Perturbations: Theoretical Protocol and Application in Systems Biology. In: Proceedings of the 2nd International Conference on Complex Intelligent and Software Intensive Systems, pp. 675–681. IEEE Computer Society Press, Los Alamitos (2008)
8. Hopfield, J.J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proceedings of the National Academy of Sciences of the United States of America 79, 2554–2558 (1982)
9. Albert, R., Barabási, A.L.: Statistical Mechanics of Complex Networks. Reviews of Modern Physics 74, 47–97 (2002)
10. Barabási, A.L., Oltvai, Z.N.: Network Biology: Understanding the Cell's Functional Organization. Nature Reviews Genetics 5, 101–113 (2004)

11. Wolfram, S.: Statistical mechanics of cellular automata. Reviews of Modern Physics 55, 601–644 (1983)
12. Ayoubi, R.A., Ziade, H.A., Bayoumi, M.A.: Fault Tolerant Hopfield Associative Memory on Torus. In: Proceedings of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, p. 369, 8 pages. IEEE Computer Society Press, Los Alamitos (2003)
13. Regnault, D., Schabanel, N., Thierry, E.: Progresses in the Analysis of Stochastic 2D Cellular Automata: A Study of Asynchronous 2D Minority. In: Kučera, L., Kučera, A. (eds.) MFCS 2007. LNCS, vol. 4708, pp. 320–332. Springer, Heidelberg (2007)
14. Demongeot, J., Sené, S.: Boundary Conditions and Phase Transitions in Neural Networks. Simulation Results. Neural Networks (to appear, 2008)
15. Withfield, M.L., Sherlock, G., Saldanha, A.J., Murray, J.I., Ball, C.A., Alexander, K.E., Matese, C., Perou, C.M., Hurt, M.M., Botstein, D.: Identification of Genes Periodically Expressed in the Human Cell Cycle and Their Expression in Tumors. Molecular Biology of the Cell 13, 1977–2000 (2002)
16. Abacci, A.: A Modelisation of the CCMRN in Superior Eukaryotes Cell Population. Master's thesis, University Joseph Fourier of Grenoble (2006)
17. Demongeot, J., Aracena, J., Thuderoz, F., Baum, T.P., Cohen, O.: Genetic Regulation Networks: Circuits, Regulons and Attractors. Comptes Rendus Biologies 326, 171–188 (2003)
18. Demongeot, J., Thuderoz, F., Baum, T.P., Berger, F., Cohen, O.: Bio-array Images Processing and Genetic Networks Modelling. Comptes Rendus Biologies 326, 487–500 (2003)
19. Aracena, J., Ben Lamine, S., Mermet, M.A., Cohen, O., Demongeot, J.: Mathematical Modelling in Genetic Networks: Relationships Between the Genetic Expression and Both Chromosomic Breakage and Positive Circuits. IEEE Transactions in Systems Man Cybernetics 3, 825–834 (2003)
20. Aracena, J., Demongeot, J., Goles, E.: Mathematical Modelling in Genetic Networks. IEEE Transaction in Neural Networks 15, 77–83 (2004)
21. Aracena, J., Demongeot, J., Goles, E.: Fixed Points and Maximal Independent Sets on AND-OR Networks. Discrete Applied Mathematics 138, 277–288 (2004)
22. Aracena, J., Demongeot, J., Goles, E.: On Limit Cycles of Monotone Functions With Symmetric Connection Graphs. Theoretical Computer Science 322, 237–244 (2004)
23. Aracena, J., Demongeot, J.: Mathematical Methods for Inferring Regulatory Networks Interactions: Application to Genetic Regulation. Acta Biotheoretica 51, 391–400 (2004)
24. Michon, F., Forest, L., Collomb, E., Demongeot, J., Dhouailly, D.: BMP-2 and BMP-7 Play Antagonistic Roles in Feather Induction. Development (in press, 2008)
25. Elena, A., Demongeot, J.: Interaction Motifs in Regulatory Networks and Structural Robustness. In: Proceeding of the International Conference on Complex, Intelligent and Software Intensive Systems, pp. 682–686. IEEE Computer Society Press, Los Alamitos (2008)
26. Demongeot, J., Moreira, A.: A Circular RNA at the Origin of Life. Journal of Theoretical Biology 249, 314–324 (2007)

Study of Signalized Intersection Crashes Using Artificial Intelligence Methods

Pei Liu[1], Shih-Huang Chen[2], and Ming-Der Yang[3]

[1] Assistant professor, Dept. of Transportation Technology & Management, Feng-Chia Univ., 100 Wen-Hwa Rd., Seatwen, TaiChung, 407 Taiwan
peiliu@fcu.edu.tw
[2] Assistant professor, Dept. of Transportation Technology & Management, Feng-Chia Univ., 100 Wen-Hwa Rd., Seatwen, TaiChung, 407 Taiwan
shc@fcu.edu.tw
[3] Professor, Dept. of Civil Engineering, National Chung Hsin Univ., 250, Kuo Kuang Rd., Taichung 402, Taiwan
mdyang@dragron.nchu.edu.tw

Abstract. High percentage of traffic crashes occurred at intersections. Generally, human error is the only cause to be blamed. However, approaching roadside environment that drivers confronted with right before crash occurrence is actually critical to crash occurrence. In this study, environmental factors critical to intersection crashes occurrence were identified via negative binomial regression and artificial neural networks. With these factors, data mining was then applied to find the rule for judging intersections safety. The 3,441 crashes occurred at 102 intersections in Taichung, Taiwan during 1999 ~ 2004 were collected. Numbers of crashes in specific approaching direction combinations were then modeled. It was found that geometry of approaching roadways was indeed critical. Total 47 safety rules generated from Gini decision tree can serve as a tool for safety evaluation of intersections. Finally, although no single factor can induce crashes alone, road width seems to be a crucial factor for intersection-related crash occurrence.

Keywords: intersection crashes, negative binomial regression, artificial neural networks, data mining, approaching direction combinations.

1 Introduction

In most developed countries, annual increase rate of motor vehicles will generally exceed that of roadways. Such a trend is also observed in Taiwan. Annual increase rate of motor vehicles and of roadways in Taiwan during 1999 to 2004 were 3.68% and 1.47%, respectively. This implies that exposure of motor vehicles, as well as traffic safety problem, increases. Actually, crash statistics does exhibit higher crash rates during that time period than ever in Taiwan. Traffic crashes generally occur at intersections in which many conflict points exist. According to the Federal Highway Administration of United States [1], intersection-related crashes make up approximately 23%

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 987–997, 2008.

of all fatal crashes, and more than half of all fatal and injured crashes occur at intersections. Considering resulted asset damage and life loss, it is clear that there is a need to understand more about intersection-related traffic crashes.

Statistical models have been widely applied to analyze traffic crashes. Most of them are linear regression models. Those models considered number of crashes or crash rate as a linear function of selected characteristics [2-4]. However, linear regression models assume that number of crashes is normally distributed. This assumption is unrealistic since traffic crashes are generally discrete, random, sparse and non-negative. To overcome this problem, Poisson regression models and the negative binominal regression models were then adopted in more recent studies. Moses and Savage [5] constructed a negative binominal regression (NBR) model to analyze truck accidents and reported acceptable results. On the other hand, Shankar, Mannering, and Barfield [6] utilized zero-inflated Poisson (ZIP) and zero-inflated negative-binomial (ZINB) models to overcome zero or sparse crash record problem.

Artificial neural network (ANN) models have been proven to be capable of modeling complicated multivariate phenomena. ANN has been applied to many areas of studies, but seldom in traffic safety [7-9]. Data mining (DM) techniques were introduced to crash-related researches not long ago. Kuhnert, Do, and Mcclure used classification and regression trees (CART), multivariate adaptive regression splines (MARS), and logistic model to analyze damage severity of crashes [10]. Similar study was performed by Wang, 2004, in her master thesis [11].

In this study, Back-propagation neural networks (BPN) were applied to model the effect of intersection characteristics on numbers of intersection-related traffic crashes. Meanwhile, NBR model was created for verification of ANN models. With these models, most sensitive variables were identified. Moreover, data mining models were adopted to further identify safety rules for intersections using these variables. Model predictions were then compared and discussed.

2 Analytical Framework

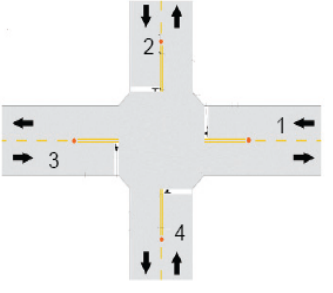
In this study, intersection characteristics were assumed to have concurrent influence on the occurrence of intersection-related traffic crashes. Human factors were excluded in proposed models. Intersection characteristics can be divided into traffic-related and geometry-related ones. Among them, traffic volume, peak hour factor, and left/right turn ratio, etc., were considered as the traffic-related characteristics. On the other hand, number and width of traffic lanes, signal cycle time, traffic signs, and types of medians, etc., were considered as the geometry-related characteristics. Details of these proposed models are described in the following sections. Nonetheless, the concept of approaching direction combinations should be elaborated first for better understanding of the basic idea of proposed models.

2.1 Approaching Direction Combinations

Vehicles involved in an intersection-related crash generally came from different directions before they ran into each other. As a result, what these drivers experienced before that crash would normally be different too. Hence, there is a need to know

which direction they came into the intersection to correctly collect corresponding geometry and traffic related characters. We thus defined the directions which involved vehicles came toward the intersection as the approaching directions. The approaching directions of vehicles involved in a crash thus form the approaching direction combination (ADC). To simplify our study, two-vehicle crashes were modeled. For crashes with more than two vehicles involved, those two vehicles with major responsibilities were considered to form ADC of that crash. On the other hand, one-car crashes and crashes with unclear responsibility were not included in this study.

An intersection connecting with N approach directions have $N(N+1)/2$ ADCs. This formula includes all cases that two vehicles coming from the same or different approaching directions. An example of calculation of ADCs for an intersection is given in figure 1. With this formula, the 102 intersections investigated in this study were decomposed into a total of 1,039 ADCs. Total numbers of crashes and characteristics corresponding to each ADC were recorded and treated as the dependent and independent variables of the proposed models. A brief summary of the distribution of ADCs with respect to numbers of crashes is illustrated in figure 2. As in figure 2, one can see that most ADCs correlated with few numbers of crashes. More precisely, 84.2% of the studied ADCs were with less than 6 crashes during studied years. This implies that certain ADCs may have the tendency to induce more crashes than the others. Consequently, the number of crashes occurred in a specific intersection were convinced to be associated with the characteristics of that intersection.



For intersection shown in the left

ADC 1 = 1, 1	ADC 6 = 2, 3
ADC 2 = 1, 2	ADC 7 = 2, 4
ADC 3 = 1, 3	ADC 8 = 3, 3
ADC 4 = 1, 4	ADC 9 = 3, 4
ADC 5 = 2, 2	ADC 10 = 4, 4

$$\Rightarrow \text{ADCs} = \frac{N(N+1)}{2} = \frac{4(4+1)}{2} = 10$$

Fig. 1. Distribution of ADCs with respect to number of crashes

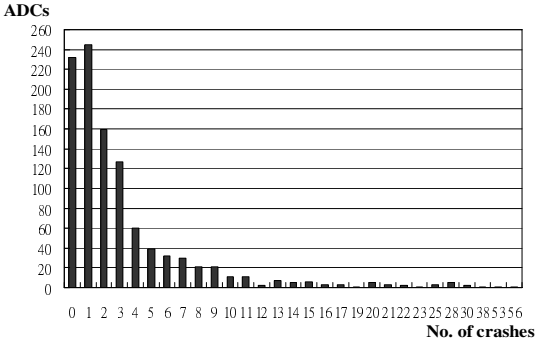


Fig. 2. Distribution of ADCs with respect to number of crashes

2.2 Variables Used in Proposed Models

There are generally two groups of independent variables, viz. traffic-related and geometry-related variables adopted in traffic crash related analyses. Originally, 9 commonly used traffic related variables including overall traffic volume, peak hour factor, percent of left turn vehicles, percent of right turn vehicles, percent of heavy vehicles, percent of motorcycles, signal cycle time, green light seconds, and left turn signal seconds; and 19 geometry related variables, namely road width, numbers of fast traffic lanes, numbers of slow traffic lanes, width of shoulder, median type, median type between fast and slow traffic lanes, number of left/right turn lanes, existence of left/right turn signals, number of signals, included angle between ADC approaches, complexity of signal timing scenario, road width/(yellow & red light seconds), existence of warning signs, integrity & visibility of traffic marking, grade, pavement distress or road surface debris, illumination, obstacles within traffic lanes, and obstacles within the intersection were taken into account.

Due to the fact that the above variables may be interrelated and thus are collinear, correlation coefficients between each pair of variables were evaluated. Those pairs of variables with Pearson correlation coefficients greater than 0.75 were first identified. Stepwise regression analysis was then applied to delete unnecessary variables. After this process, 18 variables were selected as model input variables. These identified critical variables are *numbers of traffic lanes*, *number of left turn lanes*, *road width*, *median type*, *median type between fast and slow traffic lanes*, *number of signals*, *exist. of left/right turn signals*, *complexity of signal timing scenario*, *signal cycle time*, *green light seconds*, *existence of warning signs*, *obstacles within the intersection*, *obstacles within traffic lanes*, *overall traffic volume*, *directional traffic volume*, *% of left turn vehicles*, *% of right turn vehicles*, and *% of motorcycles*.

Information from both approaching directions of each ADC had to be recorded; yet *total traffic volume* is unique for each intersection, thus there were total 35 input variables in the proposed model. These variables may be either numeric or nominal, their ranges may varied widely too. Though ANN can manipulate data with wide ranges, domains of all selected variables were converted into [–1, 1] to enhance the learning speed.

2.3 Data Collection

Total 102 signalized intersections in Taichung city, Taiwan were investigated in this study. Crash records and all input variables with respect to these intersections during year 1999 ~ 2004 were collected. Total 3,441 crashes were reported in these intersections during those years. It should to note that the most dangerous ADCs in some important intersections in this study varied with time. It may be resulted from strengthened law enforcement or local development in the vicinity of those intersections. However, this phenomenon is neglected since it's not very common. Data of official documented independent variables, such as signal timing scenarios, width of road, width of traffic lanes, etc. were provided by the Bureau of Transportation of the Taichung city government. On the other hand, those variables not listed in official records were collected from field measurements. Since traffic volume survey is time and money consuming, traffic volumes of all intersections studied were cited from

research reports published within three years to the time of study [12]. Consequently, these values were not concurrent. Nonetheless, due to the fact that traffic volume in a developed city is in somewhat stable in a period of time, and traffic volume was applied as an indication of event exposure, thus no modification on these data was performed. As to the included angles of ADCs, values were determined by performing measurement directly on electronic maps of the Taichung city using Mapinfo.

Besides the aforementioned independent variables, the dependent variable used in the proposed model was the number of intersection-related crashes with respect to each ADC. The crash records were collected with help from the Traffic Police division of the Taichung Police Bureau (TCPB). With the TCPB crash record database, information such as crash spot, date, time, illumination, and weather condition of each crash were able to be recorded directly. However, exact ADC of each crash, which could only be observed from in-situ drawings in the original crash record, was not included in the TCPB crash record database. Consequently, ADCs were obtained manually by going through all original crash reports filed during the studied time period. Moreover, all personal information listed in these reports was kept confidential and intact during this process to protect the privacy of every individual involved in all crashes studied.

2.4 Negative Binominal Regression

Negative binominal regression is a nonlinear regression model, which adopts gamma distribution to depict expected value to overcome the character of average value not equal to variance. Such formulation makes NBR suitable for traffic safety analysis. As Shankar discussed, a road section with zero crash record is not necessary a safe one. Zero crash record may be resulted from unreported crashes, or no crash occurred during the survey period. He thus suggested assign specific probability value to the zero crash case. This concept was adopted in this study, i.e. ADCs with zero crash records were all taken into account in model generation and a 0.001 probability value was assigned to the zero crash case. To construct the NBR model, 75% of the total 1039 ADCs (i.e. 779 ADCs) were selected as the learning set, rest 25% (260 ADCs) were used to verify the generated model. LIMDEP 7.0 was adopted for parameter verifications and κ^2 tests. Results of are given in Table 1.

It can be seen from table 1 that κ^2 is 43.2857, discrete parameter α is 0.7252, and P value is 0.0000 (< 0.05), thus NBR model was an acceptable model for this problem. Meanwhile, total 9 variables are significant for $\alpha = 0.05$ in the model, viz. *road width, number of signals, complexity of signal timing scenario, signal cycle time, existence of warning signs, obstacles within traffic lanes, overall traffic volume, directional traffic volume, and percentage of motorcycles*. Moreover, *obstacles within the intersection* is significant for $\alpha = 0.1$.

Model verification indicated that NBR model can give acceptable predictions (R^2 of the generated model is 0.843). However, the model tends to give under-estimations, the overall negligence rate is 57.42%, and the greatest error is -25.43. Such results may be due to the fact that more than 80% of ADCs are with low value crash records, thus under-estimation occurs for ADCs with higher value crash records.

Table 1. Summary of Negative Binominal Regression Analysis

Variables	Coeff.	P value	Variables	Coeff.	P value
Constant	-0.740	0.153	Green light seconds 1	0.001	0.766
# of traffic lanes 1	-0.053	0.105	Green light seconds 2	0.001	0.896
# of traffic lanes 2	-0.035	0.114	exist. of warning signs 1	0.347	0.004
# of left turn lanes 1	-0.001	0.995	exist. of warning signs 2	-0.177	0.140
# of left turn lanes 2	0.166	0.114	obstacles within intersection 1	-0.190	0.074
road width 1	0.017	0.014	obstacles within intersection 2	-0.051	0.627
road width 2	-0.005	0.060	obstacles in traffic lanes 1	0.015	0.876
median type 1	-0.012	0.826	obstacles in traffic lanes 2	-0.236	0.009
median type 2	0.035	0.548	overall traffic volume	0.00	0.001
median between fast / slow traffic lanes 1	0.015	0.792	directional traffic volume 1	0.0001	0.062
median between fast / slow traffic lanes 2	-0.082	0.196	directional traffic volume 2	0.0001	0.015
# of signals 1	0.196	0.023	% of left turn vehicles 1	0.126	0.617
# of signals 2	-0.186	0.026	% of left turn vehicles 2	0.118	0.647
Exist. of left/right turn signals 1	0.069	0.701	% of right turn vehicles 1	0.442	0.115
Exist. of left/right turn signals 2	0.279	0.125	% of right turn vehicles 2	-0.168	0.534
complexity of signal timing scenario 1	-0.119	0.015	% of motorcycles 1	0.069	0.067
complexity of signal timing scenario 2	0.049	0.076	% of motorcycles 2	0.076	0.048
signal cycle time	0.005	0.031	α (alpha)	0.725	0.000

Note: "1" and "2" stand for approaching direction 1 and approaching direction 2, respectively.

2.5 Back-Propagation Model

It was intended to apply ANN to associate the interrelation among all input variables and the output variable. Feed-forward, supervised ANNs are suitable for this type of problem. A back-propagation network (BPN) was thus applied. BPN is a supervised neural network which uses training set to learn embedded rule of that data set. It utilizes weights and aggregation function to do forward calculation. By checking the difference between output vector and target vector, it may go backward to adjust weights and do forward calculation again. With such back and forth iteration, BPN can find the most suitable weight vector and give accurate prediction of target vector.

In the model generation process, the learning rate m was set at 0.1. This learning rate is a parameter that determines the amount of weight change at each learning cycle, so as to optimize the convergence to the minimum MSE. An optimum value for m has to be found in order to improve accuracy, to speed up the convergence process, and to prevent the network from being trapped in a local minimum of the multi-dimensional error function. The weights were set randomly between –1.0 and 1.0 before training. Performance of the network was checked on another subset of the

input vectors different from the learning set. Performance was also monitored during the training phase in order to prevent overtraining of the network. Training phase was forced to stop when the MSE was no longer diminishing.

In our model, basic BPN with one input layer, one hidden layer, and one output layer was arranged. There were 34, 21, and 1 neuron in the input layer, hidden layer, and output layer, respectively. The sigmoid threshold transfer function was applied as the transfer function for the input and hidden layers; whereas, the linear transfer function was used between the hidden layer and output layer. Meanwhile, the Fletcher-Reeves update method, one kind of the conjugate gradient methods, was selected as the optimizing algorithm.

The model was generated using MATLAB 6.5 on an IBM compatible personal computer. As in the BNR model, 75% and 25% of ADCs were randomly assigned into learning set and verification set respectively. The best network stopped with MSE equal to 2.988×10^{-6}. The model was then applied to the verification set. The predictions were highly correlated to actual values (R = 0.995 as shown in figure 3). If we consider the model misjudges whenever the absolute error between the predicted and actual values is greater than 0.5 (before the round-off procedure), then the negligence rate is at 13.53% for these 260 ADCs. It should be noted that these misjudged cases all under or over estimated within 2 crashes. It was thus believed that the correlation between intersection characteristics and the number of intersection-related crashes can be successfully modeled by neural networks. Meanwhile, the intersection characteristics do correlate with the number of intersection-related crashes. Moreover, ANN model can give more accurate predictions than NBR model.

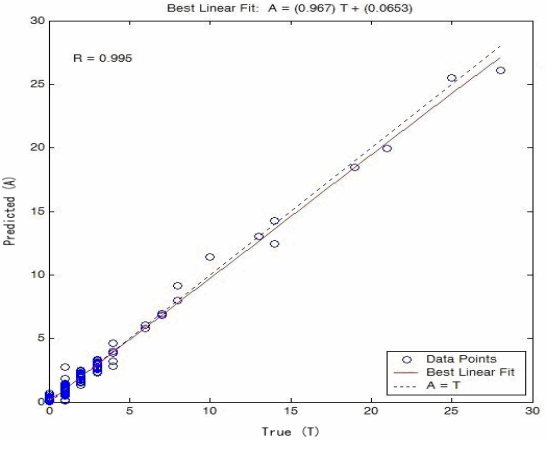


Fig. 3. Correlation between predicted and actual values for the verification set

2.6 Sensitivity Study

Variable sensitivity studies were performed to find out those variables with greater influence. Generally, sensitivity is used as a tool to examine the effect of variance of specific variable on the dependent variable. Greater sensitivity value implies the dependent variable is more sensitive to variance of that variable.

Elasticity analysis was applied for the NBR model. Direct elasticity and pseudo-elasticity were adopted for continues variables and nominal variables, respectively. Direct elasticity was performed as in eq. (1).

$$E_{X_{ij}}^{\lambda_i} = \beta_j X_{ij} \tag{1}$$

where E, X_{ij}, β_j, and λ_i are elasticity, value of variable i with respect to variable j, coefficient of X_{ij}, and expectation value of variable i, respectively. As to the pseudo-elasticity, the following equation was adopted.

$$E_{X_{ij}}^{\lambda_i} = \frac{\exp(\beta_j)-1}{\exp(\beta_j)} \tag{2}$$

Elasticity value of each variable will reveal importance of that variable to the dependent variable. Meanwhile, the sign of elasticity indicates they are positively or negatively correlated. With the above equations, elasticity values of all variables were calculated. It was found that *road width*, *overall traffic volume*, *percentage of motorcycles*, *complexity of signal timing scenario*, and *directional traffic volumes* are dominant variables. Meanwhile, traffic related variables are apparently more important than geometry related variables.

For the ANN network structure, sensitivity of the dependent variable with respect to every single variable can be derived as the following equation,

$$s_i = \sum_{i=1}^{l} \sum_{j=1}^{m} w_{ij} \frac{\partial n_j}{\partial p_i} w_{j1} \frac{\partial t}{\partial q_j} \tag{3}$$

where

s_i : sensitivity of the i^{th} variable (neuron in input layer),
l, m : number of variables in input and hidden layers, respectively,
w_{ij} : weight coefficient between i^{th} neuron in input layer & j^{th} neuron in hidden layer,
w_{j1} : weight coefficient between j^{th} neuron in hidden layer & the neuron in output layer,
n_j : output of the j^{th} variable in hidden layer,
p_i : the i^{th} variable in input layer,
q_j : the j^{th} variable in hidden layer,
t : the output variable.

With the above equation, sensitivities of all 35 input variables were calculated for each of the 779 ADCs in the learning set. In general, the dependent variable is either positively or negatively correlated to an independent variable. However, the calculated sensitivity values of every variable in the proposed model exhibited both positive and negative values. In our model, occurrence of crashes was assumed to be influenced by the associated effect of all input variables. This implies that variables do not "work" alone and thus won't always have the same effect on the output variable. Consider the variable *number of fast traffic lanes* as an example. For a jammed highway system, adding one fast traffic lane can generally release the demand, reduce stop-and-go movement, and may thus reduce crash occurrence. On the other hand, adding one fast traffic lane to a highway with "A" level of service may just induce much faster driving behavior and thus may cause more crashes. Moreover, since the

final weight coefficients obtained from a BPN model are the result of multiple forward and backward iterative calculations. They are actually influenced by the associated effect of all variables. Consequently, the unique combination of all variables in each ADC makes the sensitivity of specific variable unique in that case.

Since the sensitivity values fluctuated, a scheme similar to the least-square method was introduced to identify those variables which were most sensitive to the output value. Sensitivity values of one variable calculated for all ADCs were squared, summed, and averaged to get the averaged sensitivity value of that variable. With this definition, seven most sensible variables were found to be *road width*, *directional traffic volume*, *% of left turn vehicles, percentage of motorcycles*, *complexity of signal timing scenario*, *overall traffic volume*, and *% of right turn vehicles*.

Similar results were obtained in elasticity analysis and sensitivity analysis. Variables commonly observed from these analyses, viz. *road widths*, *% of motorcycles, complexity of signal timing scenario, overall traffic volume, % of left turn vehicle*, *signal cycle time, directional traffic volume, existence of warning signs*, *median type between fast and slow traffic lanes* and *% of right turn vehicles*, were judged as the most sensitive variables.

2.7 Data Mining Analysis

The above mentioned 10 most sensitive variables were adopted in the data mining model. Among them, *overall traffic volume* and *signal cycle time* are the same for all ADCs in an intersection, hence there are 18 variables in the CART model. The Gini classification and the Twoing classification were first adopted for model construction. Twoing classification uses bisection principle to generate children nodes from parent nodes. Whereas the Gini classification generates two children nodes based on node purities. Consequently one can reasonably predict that Gini can give better classification trees. The results confirmed such observation.

In this study, 6 output classes were specified in the Gini decision tree generation to identify intersection safety (as listed in table 2).

Table 2. Definitions of output levels specified in the Gini model

Output class	Number of crashes occurred Yearly	Dangerous level
0	0	Safe
1	1	Fairly safe
2	2	Fairly unsafe
3	3 ~ 4	Fairly dangerous
4	5 ~ 8	dangerous
5	> 9	Extremely dangerous

The generated Gini decision tree for current study has total 47 terminal nodes, which stands for 47 safety rules. The tree is huge and is only partly shown in figure 4 to illustrate the way of reading it. As shown in figure 4, the underlined class (greatest number) in each terminal node is the class represents that terminal node, e.g. terminal node 1 is judged as a safe class since C0 has the greatest number 17. One can tell

from the tree that the safety rule of terminal node 1 is road width 1 < 14.25 and AADT 2 < 633.5, which means an ADC is safe when road width of approaching direction #1 is smaller than 14.25m and annual averaged daily traffic volume of approaching direction #2 is less than 633.5. Based on this principle, the generated tree gave intersections 10 safe rules, 13 fairly safe rules, 6 fairly unsafe rules, 7 fairly dangerous rules, 7 dangerous rules, and 4 extremely dangerous rules. With the terminal nodes and associated intermediate nodes, one can study the variable combinations lead to all six safety levels. One can then further understand what kind of variable combinations should be avoided to preserve a safe intersection. In our study, most safety rules involved 4 to 7 conditions, and should not be difficult for in-situ evaluation of certain ADC of an intersection. However, it will be difficult to go through all 47 rules for all ADCs of an intersection. Hence, it may be better to identify the most dangerous ADC of an intersection using BPN model. Then check its safety with safety rules generated here.

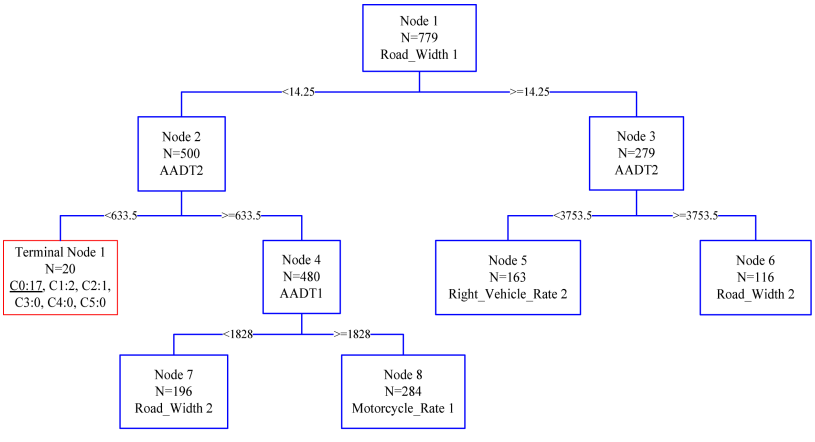


Fig. 4. The Gini decision tree generated in this study (partly shown)

3 Conclusion

In this study, negative binominal regression model and back propagation neural network model were generated using crash records and characteristics of 102 signalized intersections in Taichung city, Taiwan. It was found that BPN model can give more reliable predictions than NBR model. Variables critical to crash occurrence were identified via sensitivity study on the generated models. Finally a CART Gini decision tree was generated using the above mentioned critical variables.

Brief conclusion can be given as followed. The generated BPN model performed well with 2.98×10^{-6} MSE value, 0.994 correlation coefficient, and 13.53% misjudgment rate. This implies that artificial neural network does have the ability to manipulate multi-variate problems. The sensitivity study results indicate that the effect of specific variable on number of intersection-related crashes is not identical for various intersections. This confirms the perception that there is no single dangerous environmental factor, instead, various factors interact with each other to form a dangerous

intersection. Meanwhile, most sensitive characteristics on intersection-related crashes are found to be *road widths*, *% of motorcycles, complexity of signal timing scenario, overall traffic volume, % of left turn vehicle*, *signal cycle time, directional traffic volume, existence of warning signs*, *median type between fast and slow traffic lanes* and *% of right turn vehicles*.

As to the CART decision tree, it can serve as an evaluation tool for highway agencies. Gathering values of all variables used to generate the tree, one can evaluate specific intersection following the top-down tree structure. Moreover, one can also extract information from the tree structure to know threshold value of certain variable. With this kind of information, highway and traffic engineers can make important decisions like appropriate road width or signal cycle time for specific intersections.

References

1. National Transportation Statistics. The Bureau of Transportation Statistics, U.S.A (2007) (January 22, 2008), http://www.bts.gov/publications/national_transportation_statistics/2007
2. Lee, C., Saccomanno, F., Hellianga, B.: Analysis of Crash Precursors on Instrumented Freeways, 81st TRB CD–ROM (2002)
3. Miaou, S.P., Lum, H.: Modeling Vehicle Crashes and Highway Geometric Design Relationships. Accident Analysis and Prevention 25(6), 689–709 (1993)
4. Liu, G.X., Popoff, G.X.: Provincial–Wide Travel Speed and Traffic Safety Study in Sasketchewan. Transportation Research Record 1595, TRB, National Research Council, Washington, DC, pp. 1–8 (1997)
5. Moses, L.N., Savage, I.: The Effect of Firm Characteristics on Truck Accidents. Accident Analysis and Prevention 26(2), 173–179 (1994)
6. Shankar, V., Mannering, F., Barfield, W.: Effect of Roadway Geometrics and Environmental Factors on Rural Freeway Accident Frequencies. Accident Analysis and Prevention 27(3), 371–389 (1995)
7. Mussone, L., Ferrari, A., Oneta, M.: An analysis of urban collisions using an artificial intelligence model. Accident Analysis and Prevention 31, 705–718 (1999)
8. Wei, K.Y.: Using Collision Diagram and Neural Network to Study the Effect of Engineering Factors on Traffic Crashes, M.S. thesis, National Chen–Kung University, Taiwan (2000) (in Chinese)
9. Chou, Y.J.: The Research of Urban Crash Severity with Artificial Neural Networks, M.S. thesis, National ChenKung University, Taiwan (2001) (in Chinese)
10. Kuhnert, P.M., Do, K.A., McClure, R.: Combining non-parametric models with logistic regression: an application to motor vehicle injury data. Computational Statistics & Data Analysis 34(3), 371–386 (2000)
11. Wang, S.W.: Application of data mining techniques on evaluation of severity of traffic crashes, MS thesis, National Chia-Yi university, Taiwan (2004)
12. Taichung City Government, Rehabilitation of Dangerous Road Sections and Intersections in Taichung City: Assignment of 2001 (in Chinese), Taiwan (2002)

Fault Diagnosis of Industrial Systems with Bayesian Networks and Neural Networks

Luis E. Garza Castañon[1], Juan Pablo Nieto González[1], Mauricio A. Garza Castañon[2], and Rubén Morales-Menéndez[1]

[1] Department of Mechatronics and Automation, ITESM Monterrey Campus
Av. Eugenio Garza Sada Sur No. 2501
Monterrey, N.L. 64,489 México
legarza@itesm.mx,juan.pablo.nieto@itesm.mx,rmm@itesm.mx
[2] Corporación Mexicana de Investigación en Materiales
P.O. Box 491, Saltillo, Coah., 25000
magarza@comimsa.com

Abstract. In this work we propose a two-phases fault diagnosis framework for industrial processes and systems, which combines Bayesian networks with neural networks. The first phase, based just on discrete observed symptoms, generates a set of suspicious faulty process components. The second phase analyzes continuous data coming from sensors attached to components of this set and identifies the fault mode of each one. In first phase we use a discrete Bayesian network model, where probabilistic relationships among system's components are stated. In second phase, we analyze sensor measurements of suspicious faulty components with a probabilistic neural network, previously trained with the eigenvalues of collected data. We show promising results from simulations performed with a 24 nodes power network.

Keywords: Fault Diagnosis, Power Networks, Neural Networks, Bayesian Networks, First-order Logic.

1 Introduction

The monitoring and supervision of industrial processes and systems plays an important role to improve process performance, efficiency and product quality. Monitoring and supervision are also closely related to system safety, as it is well known that early detection of process misbehavior can help to avoid major breakdowns and incidents.

In order to tackle those problems, fault detection and system diagnosis has been a very active research domain since a few decades ago and many research works have been published. Most of the methods used are analytic, based on artificial intelligence (AI) or statistical methods [7]. Among industrial systems, a special interest has been devoted to fault diagnosis in power networks, due mainly to the high complexity and degree of interconnection present in electrical power networks, that can lead to an overwhelming array of alarms and status messages

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 998–1008, 2008.

being generated as a result of a disturbance. This can have a negative impact on the speed with which operators can respond to a contingency. Therefore, in order to increase the efficiency of diagnosis, it is necessary to use automated tools, which could help the operator to speed up the process. Very recently, the need to develop more powerful methods has been recognized, and approaches based on Bayesian networks, able to deal very efficiently with noise and modeling uncertainties have been developed in [9] and [10]. On the other hand, hybrid techniques which combine the best features of several reasoning methods, start to be used. In [8] an investigation is performed about the use of logistic regression and neural networks to classify fault causes. This paper also discusses about data insufficiency, imbalanced data constitution and threshold setting. Ren and Mi [6] propose a procedure for power systems fault diagnosis and identification based on Petri Nets and coding theory. They tested the approach with simulations over the IEEE 118-bus power system and highlight the great advantage to handle very easily future expansions. In [5] a Fault diagnosis system is presented, based on multi-agent systems. By using a negotiation mechanism between decision-making agent and a cooperative agent, fault diagnosis results can be obtained.

In this paper we propose a two-phases hybrid diagnostic framework for industrial systems, which combines Bayesian networks with neural networks. Related work to this was implemented in the past in [2], with similar approach used on the first phase and dynamic probabilistic models on second phase. Here, in the first phase, a Bayesian network model of the system is used to state probabilistic relationships between system's components. This step generates a set of suspicious faulty process components, based on discrete observed symptoms. The second phase analyzes sensor measurements data coming from components of this set and identifies the fault mode of each one. In this phase we extract relevant features from data by computing its eigenvalues. These eigenvalues are then used to train a probabilistic neural network (PNN) which in turn identifies the fault mode of the component. We show promising results from simulations performed with a power network composed by 24 nodes.

The organization of the paper is as follows: section 2 explains the fundamentals of the first phase and second phase. Section 3 gives the framework general description. Section 4 shows the case study and finally Section 5 gives the final conclusions of the paper.

2 Fundamentals

2.1 First Phase

In our diagnostic framework proposal, the first phase generates a set of suspicious faulty process components, based on discrete observed symptoms. We are using Bayesian networks as a tool to analyze the behavior of the process or system based in pure discrete observations. This phase will deliver a set of system components with higher probability of fault.

Bayesian Networks. Bayesian networks are a helpful tool to model multi-fault, multi-symptom dependency relations: every fault and every symptom is modeled by a random variable with a finite range of possible values. A graph is constructed with a node for each variable. The graph constructed has an edge from one node to another, whenever the first node models a fault directly exhibiting the symptom modeled by the second. In general, Bayesian networks are a representation for probability distributions over complex domains. Formally, we consider probability spaces defined as the set of possible assignments to some set of random variables $X_1, \ldots X_n$, each of which has a domain $Dom[X_i]$ of possible values. For example the domain of the discrete variable *Breaker_status* is $\{$*normal, open, faulted* $\}$. The domain of the variable *Voltage* can be $\Re$, or it can be discretized into some appropriate partition. The goal is to represent a joint probability distribution over these variables.

Formally, a Bayesian network is defined by a directed acyclic graph together with a local probabilistic model for each node. There is a node in the graph for each random variable $X_1, \ldots, X_n$. The edges in the graph denote direct dependency of a variable X_i on its parents $Parents(X_i)$. The graphical structure encodes a set of conditional independence assumptions (*each node X_i is conditionally independent of its non-descendants given its parents*). The qualitative independence assumptions implied by the network structure, combined with the Conditional Probability Distributions (CPDs) associated with the nodes, are enough to specify a full joint distribution through the following equation, known as the *chain rule for Bayesian networks*:

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i | Parents(X_i)). \tag{1}$$

In discrete networks, an explicit description of the joint distribution requires a number of parameters that is exponential in n, the number of variables. Bayesian networks derive their power from the ability to represent conditional independencies among variables, which allows them to take advantage of the "locality" of causal influences [4]. A variable is independent of its indirect causal influences given its direct causal influences.

In summary, with the Bayesian network model establishing known relationships between components of the system, we generate diagnoses consistent with discrete observations of the system. Every diagnosis has an associated probability of occurrence and contains a list of suspicious faulty components. The output of the first phase is the set of suspicious faulty components extracted from the intersection of most likely diagnoses. These components are going to be analyzed by second phase to determine if they are in a faulted state.

2.2 Second Phase

This phase analyzes sensor measurements data coming from the set of suspicious components delivered by first phase and identifies the fault mode of each one. In

this step we extract relevant features from sensors data by computing its eigenvalues. These eigenvalues are then used to train a probabilistic neural network (PNN) which in turn will identify the fault mode of the component.

Probabilistic Neural Networks. PNNs are conceptually similar to K-Nearest Neighbor (KNN) models [1]. The basic idea is that a predicted value of an item is likely to be about the same as other items that have close values of the predictor variables.

The nearest neighbor classification will depend on how many neighboring items are considered. If 1-NN is used, then only the closest item is considered, if 9-NN classification is used, the closest 9 items are considered, and so on. A probabilistic neural network builds on this foundation and generalizes it to consider all of the other items. The distance is computed from the item being evaluated to each of the other items, and a radial basis function (RBF) (also called a kernel function) is applied to the distance to compute the weight (influence) for each item. The radial basis function is so named because the radius distance is the argument to the function. The further some other item is from the new item, the less influence it has. Different types of radial basis functions could be used, but the most common is the Gaussian function. The PNN architecture is shown in Figure 1. The model has two layers: radial basis layer and competitive layer.

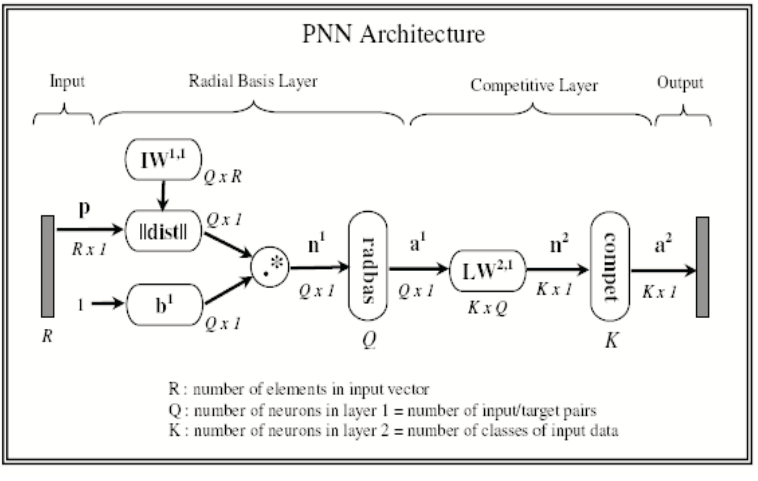


Fig. 1. PNN architecture

There are Q input vector/target vector pairs. Each target vector has K elements. One of these element is 1 and the rest is 0. Thus, each input vector is associated with one of K classes. The first-layer input weights, $IW^{1,1}$ are set to the transpose of the matrix formed from the Q training pairs, P'. When an input is presented the $||dist||$ box produces a vector whose elements indicate how close the input is to the vectors of the training set. These elements are multiplied,

element by element, by the bias and sent the *radbas* transfer function. An input vector close to a training vector is represented by a number close to 1 in the output vector a^1. If an input is close to several training vectors of a single class, it is represented by several elements of a^1 that are close to 1. The second-layer weights, $LW^{2,1}$, are set to the matrix T of target vectors. Each vector has 1 only in the row associated with that particular class of input, and 0's elsewhere. The multiplication $T \times a^1$ sums the elements of a^1 due to each of the K input classes. Finally, the second-layer transfer function, *compete*, produces a 1 corresponding to the largest element of n^2, and 0's elsewhere. Thus, the network has classified the input vector into a specific one of K classes because that class had the maximum probability of being correct.

Correlation Matrix and Eigenvalues Definitions. Following definitions are taken from [3]:

Definiton 1. A Correlation matrix describes correlation among M variables. It is a square symmetrical $M \times M$ matrix with the (ik)th element equal to the correlation coefficient r_{ik} between the (i)th and the (k)th variable. The correlation coefficient is obtained as:

$$r_{ik} = \frac{\sum_{j=1}^{n}(x_{ji} - \bar{x}_1)(x_{jk} - \bar{x}_k)}{\sqrt{\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)^2}\sqrt{\sum_{j=1}^{n}(x_{jk} - \bar{x}_k)^2}} \tag{2}$$

The diagonal elements (correlations of variables with themselves) are always equal to 1.

Definition 2. Let A be a $k \times k$ square matrix and I be the $k \times k$ identity matrix. Then the scalars $\lambda_1, \lambda_2, ..., \lambda_k$ satisfying the polynomial equation:

$$|A - \lambda I| = 0 \tag{3}$$

are called the eigenvalues or characteristic roots of a matrix A. The equation is called the characteristic equation, thus similar matrices to A and its transpose matrix have the same eigenvalues.

3 Framework Description

The proposed fault diagnosis framework is shown in figure 2.

The first step is to obtain discrete observations from the system. These observations are related to devices which will reflect in some way the status of system's components. We assume that every device has a finite set of states indicating normal or faulty operation. There is also a possibility that information related to some devices is missing due to communications problems or other abnormal situations. With the known data of every device, the first phase of diagnosis will generate a set of suspicious components. The inference task will be accomplished by applying two methods over the Bayesian network model: a

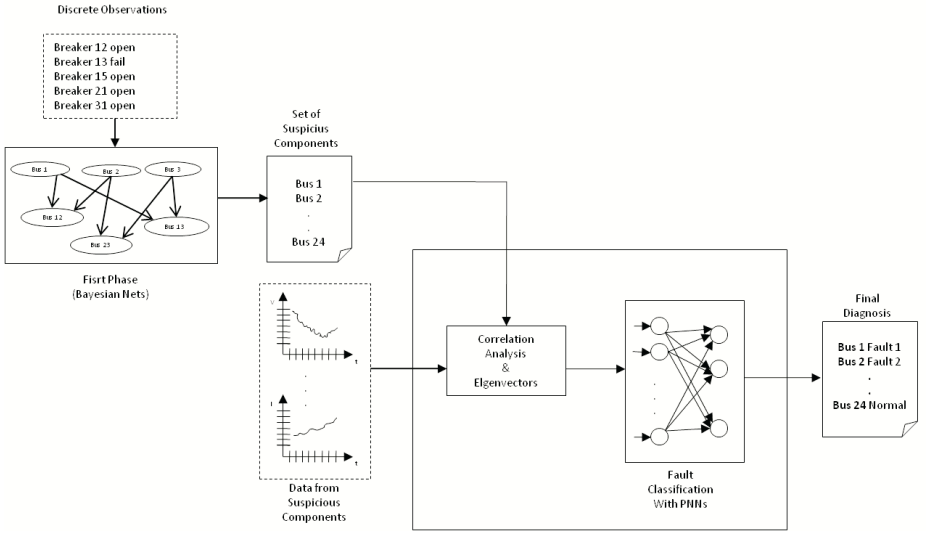


Fig. 2. General fault detection framework

first-order logic theorem prover and a likelihood weighting sampling strategy. The first inference method is related to the implementation of Bayesian network with a set of first-order logic rules. Basically, the diagnoses are generated as the explanations consistent with a set of given observations and known facts of the system. The second inference method relies on fixing observed variables, sampling non-evidence variables according to given probabilities, and weighting each sample by the likelihood of it according to the observations. The output of first phase is a list of possible diagnoses consistent with discrete observations. Every diagnosis has a probability of occurrence and contains the suspected faulted components. To extract the set of components with higher probability of failure, we use heuristics, such as the intersection of all generated diagnoses, that is the components appearing in every diagnosis.

In second phase, continuous data coming from suspicious faulty components is collected and analyzed. First, features are extracted from samples by obtaining a correlation matrix. The matrices are formed by windows of m samples and n system's components. Such matrices are constructed with normal and faulty data operation. Once we have computed the correlation matrix, we obtain their corresponding eigenvalues. In this way we have a signature of the different possible states or fault types of the system's components. The eigenvalues of the correlation matrix then serve as the input vectors of the PNN. The output from the PNN will discriminate between system's components states and give us the most likely one. In order to confirm the diagnosis, we take each sample of each component and obtain its magnitude, then a comparison is made against the constant magnitude of the two probably signature faults. This comparison serves as

a classifier that gives us the real node state and then we can use it also to locate the period of time or sample number where the fault occurs.

4 Case Study

This section shows the performance of our framework in scenarios where simultaneous multiple fault simulations were carried out. The system under testing is shown in figure 3. This network represents an electrical power system with dynamic load changes. The system has 24 components or buses, and every bus is protected by several breakers devices, which will be active when a malfunction is detected. For every bus on the system, there is a continuous voltage measurement, and for every breaker there is a discrete signal of the state of the device. We have run 24 fault simulations, including symmetrical and unsymmetrical faults at random nodes (3,9,10 and 13). Examples of simulated faults are: one line to ground (A GND), two lines to ground (A-B GND), three lines to ground (A-B-C GND), or faults between two lines (A-B or B-C). We also included data where none of the nodes of the system were under a faulty situation (NO FAULT). For every fault scenario, we also simulated the discrete response of breakers associated to faulty nodes.

A portion of the Bayesian network model for the power system, used to state the dependency between the elements of the network, is shown in figure 4. The random variables without parents are the buses or nodes, and for simplicity, we assume that they are the only source of faults. The breakers have two parents (buses), because they are the main protection for one bus and the backup protection for another bus. This scheme of backup protection allows the isolation of a bus fault, even in the case of a malfunction in the main breaker. Two different inference engines associated to the Bayesian network were used. The first one is a logic theorem prover, which in turn generates diagnoses consistent with discrete observations, and the second inference engine was based on a likelihood weighting sampling strategy. Both methods delivered accurate results in all simulated scenarios, delivering the correct set of faulted nodes, but the sampling strategy made inferences faster. In all simulations, probabilities used to indicate a fault in root nodes were set to the value of 0.001, and conditional probabilities for protection breakers were similar to those shown in table 1.

The methodology proposed is applied as follows:

1. **First Phase**
   (a) Collect discrete observations from power network protection breakers. Every breaker can be at one of three states: **OPEN**, which means that breaker responds to a faulty situation on a neighbor node; **NORMAL**, which means normal operation and breaker remains closed; and **UNKNOWN**, which means that no information has been received about breaker's status.
   (b) Run the inference engine of the Bayesian network model of the system with evidence given by the discrete observations collected above. Obtain a set of suspected faulty nodes of the network.

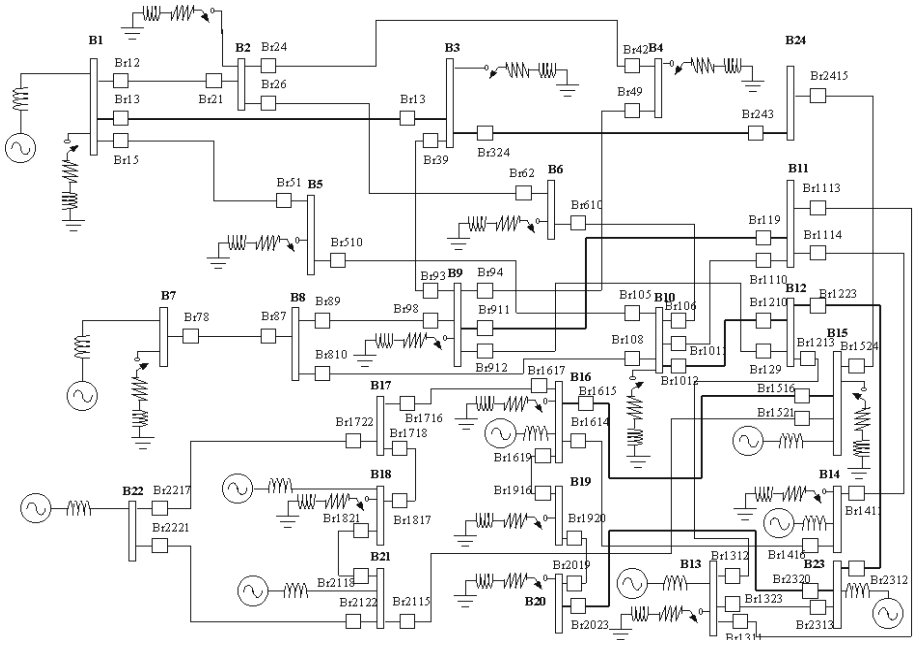


Fig. 3. Power network test system

Table 1. Conditional probabilities for protection breaker *Br12*

Bus State \ Breaker State	*NORMAL*	*OPEN*	*FAIL*
Bus 1=OK,Bus2=OK	1	0	0
Bus 1=OK,Bus2=FAULTED	0.999	0.001	0
Bus 1=FAULTED,Bus2=OK	0	0.999	0.001
Bus 1=FAULTED,Bus2=FAULTED	0	0.999	0.001

2. **Second Phase**
   (a) Obtain windows of 100 samples from normal and faulted operation history process data (electrical voltage in each node's line).
   (b) Obtain correlation matrix for each node, which gives a 3×3 matrix.
   (c) Obtain the eigenvalues from the correlation matrix (this gives 3 eigenvalues), with this 3 eigenvalues form an input vector to train the PNN.
   (d) Take a probe data set of 100 samples from the 3 voltages of every suspected node in the network.
   (e) Obtain correlation matrix for each node, which gives a 3×3 matrix.
   (f) Obtain the eigenvalues from the correlation matrix (this gives 3 eigenvalues), with this 3 eigenvalues form an input vector to the PNN.
   (g) Take the output of the PNN as one of the two probably states of the node monitored (NORMAL or FAULTY).
   (h) Take each sample of each node monitored and obtain its magnitude, then compare it against the constant magnitude of the two probably signature faults and classify

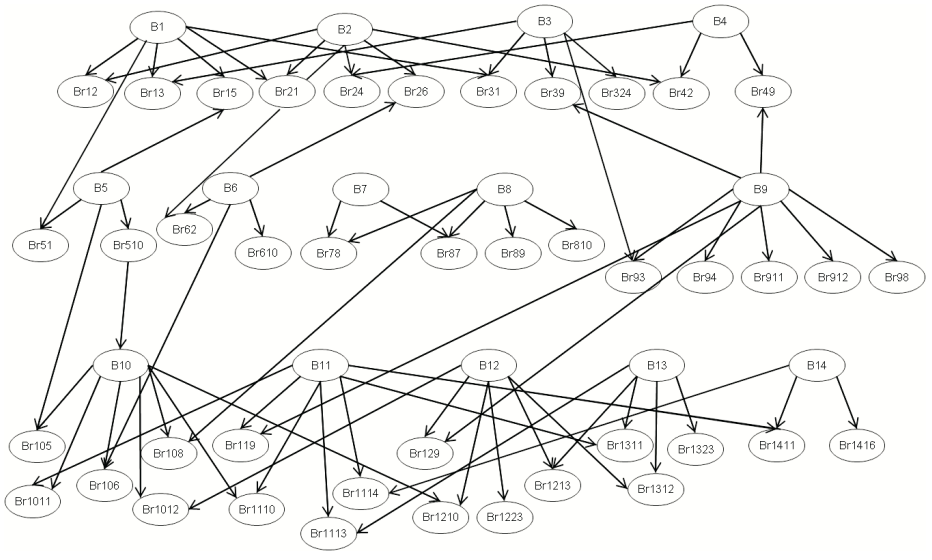


Fig. 4. A portion of Bayesian network model for the power network

it using this simple criteria. Locate the period of time or sample number where the fault occurs.

(i) Give the diagnosis of each node being monitored. If a fault is present in a specific node give the type and location of it, else print NO FAULT.

As we noticed that the first phase delivered always the correct set of faulted nodes, even in noisy scenarios, we focused the performance evaluation on the second phase, and we considered three different cases:

- Case 1, system is working properly during the first 25 samples from a total of 100, that means 25 samples are ok and 75 samples corresponds to a fault present in the system.
- Case 2 takes 50 samples of normal operation data and 50 samples with a fault present.
- Case 3 takes 75 samples of normal operation and 25 with a fault present.

Tables 2 and 3 give a summary of the obtained percentages for each of the three cases considered, according to the type of fault in table 2, and according to the node in table 3. The degradation observed on diagnosis performance is mainly due to the similarity of eigenvalues, when there are more normal operation data than faulted in a sample window. The more normal operation data in sample window the more difficult to classify eigenvalues by PNN because they are very similar.

We also have found that eigenvalues corresponding to the **A-B-C GND** fault, were very similar to those corresponding to the **NO FAULT** mode, in the same

way eigenvalues corresponding to **A GND** fault and **A-B GND** fault, as well as **A-B** fault and **B-C** fault were very similar respectively, making mandatory to implement a confirmation of the diagnosis of the system. We also show in both tables a comparison against similar work presented in [2]. Making the comparison between columns 1 and 4 of both tables, we can observe an improvement on the correct identification of most type of faults, but unfortunately there is also an increase on false positives.

Table 2. Efficiency evaluation by type of fault (in %)

Fault Type	*Case 1*	*Case 2*	*Case 3*	*Approach in [2]*
A-B-C-GND	100	100	100	100
A-B-GND	100	100	100	100
A-GND	100	85.7	92.9	85.7
A-B	100	83.3	50	83.3
B-C	100	68.8	68.8	100
NO FAULT	54.2	58.3	79.2	70.8

Table 3. Efficiency evaluation by node (in %)

Node Number	*Case 1*	*Case 2*	*Case 3*	*Approach in [2]*
3	83.3	83.3	83.3	79.1
9	79.2	75	70.8	87.5
10	91.2	87.5	62.5	87.5
13	100	95.8	100	95.8

5 Conclusions

This paper has presented a fault detection framework for industrial systems, based on Bayesian networks and Probabilistic Neural Networks. Our approach uses two phases : the first phase, based on Bayesian networks, analyze discrete observations of system's components and delivers a set of suspected faulty components. The second phase takes continuous data from these suspicious components, computes the eigenvalues of the correlation matrix and use them as inputs for a PNN to classify the node's component state. When eigenvalues are very similar, we use a confirmation of diagnosis step where we apply a simple comparison of each sample magnitude to the constant value of a certain signature fault. This step give us also the location of the fault. The experiments have shown that, when sample of continuous data contains more faulty than normal data, the diagnostic system has a better performance because eigenvalues are easily classified by the PNN as they have very different values. In future work we will address the problem of minimize the occurrence of false positives.

References

1. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. (2001)
2. Garza, L.: Hybrid Systems Fault Diagnosis With a Probabilistic Logic Reasoning Framework. PhD Thesis Dissertation, Intituto Tecnológico de Monterrey (2001)
3. Johnson, R., Wichern, D.: Applied Multivariate Statistical Analysis, 5th edn. Prentice Hall, Englewood Cliffs (2001)
4. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo (1988)
5. Peng, M., Hanli, W., Bin, C.: Study of Fault Diagnosis for Power Network based on MAS. In: 2006 International Conference on Power System Technology (2006)
6. Ren, H., Mi, Z.: Power System Fault Diagnosis Modeling Techniques based on Encoded Petri Nets. In: 2006 IEEE Power Engineering Society General Meeting (2006)
7. Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.: A Review of Process Fault Detection and Diagnosis part I, part II and part III. Computers and Chemical engineering 27, 293–311 (2003)
8. Xu, L., Chow, M.: Power distribution systems fault case identification using logistic regression ans artificial neural networks. In: Proceedings of the 13th Intl. Conf. on Intelligent Systems Application to Power Systems 2005, pp. 163–168 (2005)
9. Yongli, Z., Limin, H., Jinling, L.: Bayesian Networks-based Approach for Power Systems Fault Diagnosis. IEEE Transactions on Power Delivery 2005 21(2) (April 2006)
10. Zhao, W., Bai, X., Ding, J., Jinling, L.: Bayesian Networks-based Approach for Power Systems Fault Diagnosis. IEEE Transactions on Power Delivery 2005 21(2) (April 2006)

A Comparison between Back Propagation and the Maximum Sensibility Neural Network to Surface Roughness Prediction in Machining of Titanium (Ti 6Al 4V) Alloy

I. Escamilla[1,2], L. Torres[1], P. Perez[1], and P. Zambrano[2]

[1] Corporación Mexicana de Investigación en Materiales
Ciencia y Tecnología # 790, Frac. Saltillo 400
Saltillo, Coahuila, México
{indiraescamilla,ltorres,pperez}@comimsa.com
[2] Facultad de Ingeniería Mecánica y Eléctrica.
Ave. Universidad s/n
San Nicolás de los Garza, N. L.
{indiraepzambran}@fime.uanl.mx

Abstract. Titanium alloys are attractive materials due to their unique high strength, excellent performance at elevated temperatures and exceptional resistance to corrosion. The aerospace and military industries are the main users of this material. Titanium alloys are classified as materials difficult to machine. The correct parameters for machining are a hard to determine, and today researches are looking to develop new models to predict and optimize these parameters. The surface roughness (Ra) in turning of a titanium alloy machining Ti 6Al 4V predicted using neural and maximum sensitivity network is shown. The machining tests were carried out using PVD (TiAIN) coated carbide inserts under different cutting conditions. Confidence intervals were estimated in the model to get correct results. There are various machining parameters and they have an effect on the surface roughness. A set of initial parameters in finished turning of Ti 6Al 4V obtained from literature have been used. These parameters are cutting speed, feed rate and depth of cut. This paper shows the results obtained using these neural networks approaches to analyze the variables to model the machining process.

Keywords: Neural Network, Machining parameters, Surface Roughness, Maximum Sensibility, Back Propagation.

1 Introduction

Determining and optimizing the parameters involve in a machining process is a critical and very important task. The surface quality is one of the most specified customer requirements. It is a characteristic that could influence the performance of mechanical parts and production costs. The final surface is one of the most important considerations to determine and improve the machinability of materials. Nowadays greater attention is paid to accuracy and surface roughness of product by industry [1]. The

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 1009–1019, 2008.

machining process is a dynamical system where many variables interact, due to this, predicting and optimizing these variables are two important strategies in manufacturing. Surface roughness and dimensional accuracy are critical factors in predicting machining performance of any machining operation. The predictive modeling of machining operations requires detailed prediction of all boundary conditions. Surface roughness prediction models generally are based on laboratory experiments because is very difficult, in practice, to keep all factors under control as required to obtain reproducible results [2]. In the last two decades, the intelligent systems composed by Neural Networks, Fuzzy Logic, Evolution Computation, sometimes alone and others times in combination have been applied in manufacturing have been the subject of extensive research. This technology has become an important computing tool to solve engineering problems [3], [6]. It has led to increased research on a wide variety of industrial applications, such as product manufacturability control, process planning, etc. New research and developments are appearing en the state of art use of intelligent systems to control and optimize the machining giving support to predictions and improvement of the surface roughness at different materials [4]. The Artificial neural networks (ANNs) are one of the most powerful computer modeling techniques, based on statistical approach, currently being used in many fields of engineering to simulate the complex relationships which are difficult to describe with physical models. ANNs have been extensively applied in modeling many metal-cutting operations such as turning, milling, and drilling [5]. An Intelligent System is proposed in this research to predict and compare *back propagation and the maximum sensibility neural network to surface roughness prediction in machining of Titanium (Ti 6Al 4V) alloy.* The results shown that, this Intelligent System could serve as support in decision making to improve the machining process and productivity and to save on the cost of new products.

This research is organized in the following way: In section 1 a brief introduction will be given and section 2 has a short summary of previous research using intelligent systems in the *machining process.* The basis for research tools, Neural Network and Optimization Algorithms are shown in section 3, experimentation and results are shown in section 4, the research conclusions and future work are given in sections 5 and 6.

2 Literature Review

Ramesh in his article "Modeling for prediction of surface roughness in machining of Ti64 alloy using response surface methodology" [7] proposes a prediction model in which he includes parameters such as progress, speed and depth of cut to see their effects turning the titanium and getting the surface quality parameters response. On the other hand Che-Haron [8] worked on an investigation that determined the impact machining of Ti64 on the surface finish, checked alterations obtained in machining analyzed material characterizations in the material piece and diverse types of tooling used in the study. Rico [5] uses the Surface Response methodology and neural networks to predict the roughness. The author develops a model for predicting temperature and roughness of the cutting tool for machining steel 1018. Pawadea [4] shows in

his article titled "Effect of machining and cutting edge geometry parameters on surface integrity of high-speed turned Inconel 718" high-speed cutting and low feed rate, as well as the moderate depth of cut through the use delicate angles of cut that can ensure the generation of residual compression efforts when machining. A. Molinari [9] were devoted to comprehensive studies of chip produced during the milling Ti-6Al-4V, analyzing the process of orthogonal cutting produced at different speeds and the transformation of adiabatic shear banding. He found that the lower speeds chip becomes rougher; this due to thermomechanical limitation, which generates adiabatic shear banding, and behaves differently at high speeds. Krain [10] evaluated the effects of the change in feed rate and the thickness of the chip by changing the radius of the cutting tool as well as material and geometry of this phenomenon and its impact on the life and wear of the tool in the milling of Inconel 718. He showed that not single material or geometry gives the best results. Kopac et al. [11] used an experimental Taguchi design to determine the optimal machining parameters for best surface finish in a traditional turning. The design of Taguchi method was used to identify the impact of various parameters and how a combination of them helped to control the variation. They found that the roughness increases with increasing cutting speed. Ocktem [12] developed a model for determining the best parameters for the optimum roughness in the milling of the faces of a mold to producing a piece by means of biomechanics application (Ortez) using neural networks and genetic algorithms. It appears that a considerable amount of work is being done in the optimization of machining parameters, based on different criteria such as tool wear, vibrations, surface roughness, unit cost, etc [3] [10] [11]. Nowadays artificial intelligence (AI) based on modeling is a new trend in modeling for machining operations [2]. It was found that use of heuristic methods to model prediction of surface roughness was very limited, so emphasis was laid on the development of a surface roughness prediction model.

3 Artificial Neural Networks

Neural networks are computational paradigms that simulate some of the human brain properties, such as some rational capacities, like association, recognition of shapes and behavior patterns. Prediction is an important property of neural networks. Neural networks are non-linear mapping systems that consist of simple processors called neurons, linked it by weighted connections. Each neuron has inputs and generates an output that is the result of the information that was stored and of the processes in the hidden layers. The output signal of a neuron is fed to other neurons as input signals via interconnections. Since the capability of a single neuron is limited, complex functions can be formed by connecting many neurons. It is widely reported that structure of neural network, representation of data, normalization of inputs/outputs and appropriate selection of activation functions have strong influence on the effectiveness and performance of the trained neural network [4], [13].

Some merits of ANN applications can be summarized as, Fault tolerance and adaptability; Data-driven nature; Noise suppression capabilities

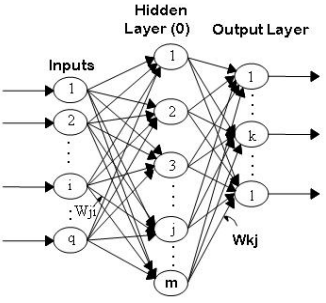


Fig. 1. Typical multi-layered feed-forward ANN

There are several neural networks [2], [4], [3], [10], This researches shown that a perceptron with back propagation learning rule is the most used to predict parameters. To develop this research a multilayer perceptron with back propagation learning rule was used. The Back Propagation Neural Network (BPNN) consists of three layers of neurons: input layer, hidden layer, and output layer Fig. 1. The input layer receives external information such as adjustable process parameters in Table 1. The output layer transmits the data and thus corresponds to various individual outputs. In this study, the number of neurons in the output layer was only one. The BPNN also incorporates hidden layers of neurons that do not interact with the outside world, but assists in performing a nonlinear feature extraction on the data provided by the input and output layers. Here, the number of the hidden layer was obtaining with design of experiment. [19]. Back propagation (BP) is based on searching a surface errors (error as a function of ANN weights) using gradient descent for points with minimum error. Each interaction BP constitutes 2 sweeps: forward activation to produce a solution and a backward propagation of the computed error to modify the weights. This starts at the input layer where each input node transmits the value received forward to each hidden node in the hidden layer. The collective effect on each of the hidden nodes is summed up by performing the dot product of all values of input nodes and their corresponding interconnection weights, as describe in the equation (1).

$$y = \begin{cases} 1, & if \sum_{i=1}^{n} w_i x_i \geq b, \\ 0, & if \sum_{i=1}^{n} w_i x_i < b, \end{cases} \tag{1}$$

Once the net effect at one hidden node is determined, the activation at that node is calculated using a transfer function (e.g. sigmoidal function) to yield an output between 0 and +1. The amount of activation obtained represents the new signal that is to be transferred forward to the subsequent layer. The same procedure of calculating the net effect is repeated for each hidden node and for all hidden layer. The net effects calculated at the output nodes are consequently transformed into activations using a transfer function. The activations are just calculated at the output node [20]. The Neural Network Maximum Sensibility (NNMS) has three layers. Every layer has a specific function. The first layer is composed by a set of source nodes and these neurons detect and distribute the input signal to the first inner layer of neurons. These

inner neurons can become active with any input signal and send a fired signal to the second inner layer. Every neuron of the first inner layer has a Gaussian activation function (2). The objective is to determine the magnitude of the distance between the weights of the inner neurons and the input signals. The input signal is recognized when there is little distance; however, the level of activation must be high.

$$G(x, \lambda, cm) = e^{-((x-cm)/\lambda)^2} \tag{2}$$

NNMS works with normalized real numbers; that means, a number in a range between 0 and 1, so weight initialization at zero value and a value is avoided and a value of -1 is used instead. The best neuron is determined using *maxp* procedure where the maximum value and its position are calculated by means of an input vector (*SN)*. [17]. In the learning procedure, the maximum sensibility state establishes the type of learning required. Use activation algorithm first. Here, a winning neuron *wn* is determined along with its activation value *v*. Locate a useless neuron, position nn. The value vnn is not considered: $(nn, vnn) = \text{minp}(\text{UW})$. If $v > ms$ then for each i input assign (3): All the neurons of the first inner layer can be activated. The best neuron is determined by selecting the most active neuron. This neuron must surpass a threshold called margin of sensibility (ms).

$$W_{wn,i} = \frac{\left(W_{wn,i} + X_i\right)}{2}. \tag{3}$$

3.1 Surface Roughness

In everyday life as well as in industry, the degree of roughness of a surface is very important. Sometimes it is necessary to have very high values of roughness, other times it is undesirable because the surface of the product requires a better appearance, or it requires the lowest surface friction because it is in contact with another surface, in this manner minimizing the phenomenon of wear materials. Surface roughness is the set of irregularities on the actual surface, conventionally defined within a section where the shape and undulation errors have been eliminated. [18] To measure the roughness of the parts an electronic instruments sensitivity micrometer called roughness meter is used to quickly determine the roughness of surfaces. There are several parameters that reflect the measurement of roughness, such as Ra, Ry, Rz. The most common is that Ra is the arithmetic mean [18] of the absolute values of the distance profile roughness of the line of the length measurement see Figure 2.

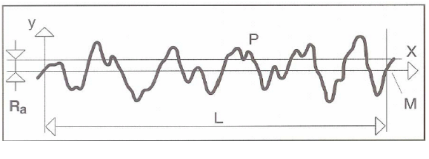


Fig. 2. Graphic Ra for measuring the roughness

4 Development of the Intelligent System

The methodology followed in this study is shown below

- Performance of the test or data extraction
- Classification of the data
- Intelligent Systems Analysis
  - Neural Network for predict
  - Maximum Sensibility Network to predict
  - Compare the results and select the best of them

The identification and optimization of variables involved in a manufacturing process is not an easy task. New research using intelligent systems in manufacturing process show that integrating the knowledge of experts into these tasks is accomplished by having good results, and to achieve this difficult feat, the global trend has been to adapt intelligent systems that are able to develop the ability to learn from these experts and their improvements.

4.1 Description Problem

The main purpose of this study is to identify the parameters that we are going to use in machining, predict the roughness produced by them, in a process of machining of titanium 64 and ensure appropriate values in the machining to guarantee the desired roughness when machining this alloy. The parameters used to carry out this analysis are shown in the table 1.

Table 1. Machining parameters used in the test

Conditions	*Units*	*Level 1*	*Level 2*	*Level 3*
Cutting Speed	m/min	40	60	80
Feed Rate	mm/rev	0.130	0.179	0.220
Depth of Cut	mm	0.500	0.750	1.000

The machining tests were extracted from Ramesh´s article [7]. The machining processes are carried out on a NAGMATI-175 lathe, Figure 3. The insert tool used was carbide cover-PVD TiAlN. The piece of work was a bar 38mm in diameter with 125mm length made of a titanium alloy (Ti-6Al-4V) Annelead. The turning took place over a length of 110mm for each experiment. The roughness was measure with a "Taylor Hobson Surtronic 3 +"surface roughness meter, shown in Figure 4. The measurements were made on a granite table, and using pattern blocks to level the pieces.

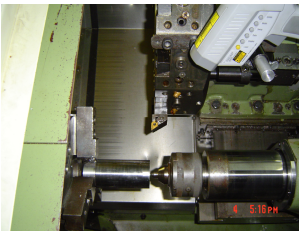


Fig. 3. Lathe NAGMATI-175

Fig. 4. Surface roughness meter

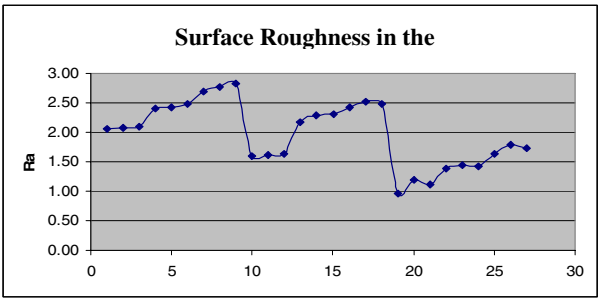


Fig. 5. Surface roughness during machining

The information used to train the neural network and maximum sensibility are showed in the figure 5.

4.1.1 Neural Network Back Propagation

To build the network is important to identify the following parameters: (1) The set of training patterns, input and target (2) A value for learning rate (3) A criterion that terminates the algorithm (4) A methodology to update weights (5) The nonlinearity function (6) Initial weight values (7) Learning moments. To develop this research a multilayer perceptron with back propagation learning rule was used. Some of variables used were: Tinp = Neurons of the input layer and bias; Tmid = Neurons of the hidden layer; Tout = Neurons of the output layer; eta = Constant learning; alpha = Moment

The parameters values were obtained by a design of experiment. The following Table 2 shows the best results obtained in the tests to train the network.

Table 2. Results with the best values for the variables used in the Neural Network

Tinp	Tmid	Tout	eta	alpha	Ntepochs	error4
3+1	25	1	0.600	0.250	5000	0.0025

Input parameters: Speed, feed and depth cut; Output value: Surface roughness value.

After make different test to train the network, the best configuration graphic, the results between experimental data and training data using the best neural network parameters are shown in Fig. 6. The results show that the network output data is reliable and close to the real data.

4.1.2 Neural Network of Maximum Sensibility (NNMS)

The neural network of maximum sensibility is motivated by the theory of functional systems where by neurons depend on the maximum sensibility generated by a specific stimulus. Only one neuron provides its response. Otherwise, when the stimulus is unknown, the activation of all neurons is required to set a response. This Network was inspired by the biological theory of functional systems. This theory was proposed and

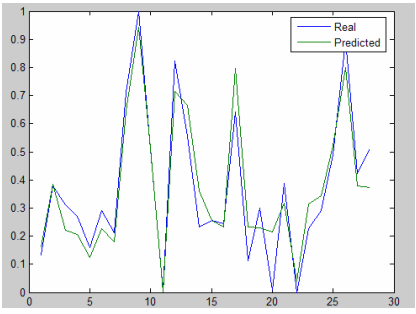


Fig. 6. Graphic real values of the experimentation vs. trained obtain of Neural Network Back Propagation

developed by neurophysiologist Pior K. Anokhin [13], [14], [15]. The functional system promotes the integration of neural information by an afferent synthesis from a dominant motivation, the environment and the memory. This afferent synthesis makes a decision to select a particular action taking into account the maximum sensibility of a dedicated structure to represent a motivation, one aspect of the environment or the memory [16], [17]. Some of variables used are: NTI = Input Neurons; NTO = Output Neurons; lambda = Neuron Sensibility; NTN = Total Neurons; ms= margin of sensibility. "NTI" and "NTO" depend on the data. They can be considered constant because the input parameters are speed, feed and depth cut. For output value the surface roughness was evaluated. With "lambda", "NTN" and "ms" we made a design of experiment to find the best results taking into account 2 levels in each factor and obtaining a model and optimizing it. The results for the best parameters for training the maximum sensibility neural network shows in the table # 3, and figure 7 shows the network output data evaluated against the real data.

Table 3. The best values for the variables used in the Neural Network of Maxim Sensibility

NTI	NTO	lambda	NTN	ms
3	1	0.39	27	0.92

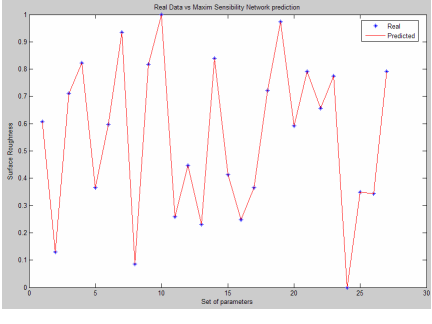


Fig. 7. Real values of the experimentation vs. trained obtain of Neural Network of Maxim Sensibility

The next step was to calculate the values from each network and match up the results to the original values. This evaluation is shown in figure # 8. The result is that both networks give results very near the real values, but the Maximum Sensibility is more accurate.

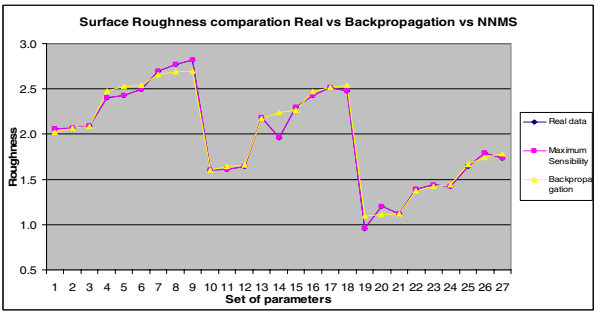


Fig. 8. Compare between Real Value vs. Backpropagation and NNMS results

4.2 Discussion

After that confidence intervals are calculated using the following formula $\bar{x} \pm \left(\left(t_{r-1,\alpha/2} \right) * \left(\frac{\sigma}{\sqrt{r}} \right) \right)$ to predict surface roughness with different parameters to use in the experiment, in the ranges in which the tests were run, in order to obtain improved parameters. Some examples are shown the Table 4 for both networks.

Table 4. Some prediction of roughness with confidence intervals to backpropagation and NNMS

Back propagation				
cutting speed	feed rate	depth cut	roughness	confidence intervals
40.000	0.130	0.500	2.066	2.075-2.050
40.000	0.130	0.600	2.062	2.071-2.047
40.000	0.130	0.700	2.060	2.069-2.045
40.000	0.130	0.800	2.066	2.075-2.051
40.000	0.130	0.900	2.081	2.090-2.065
40.000	0.130	1.000	2.099	2.109-2.084
40.000	0.140	0.500	2.125	2.135-2.110

Maxim Sensibility				
cutting speed	feed rate	depth cut	roughness	confidence intervals
40.000	0.130	0.500	2.060	2.096-2.048
40.000	0.130	0.600	2.017	2.053-2.004
40.000	0.130	0.700	2.059	2.095-2.047
40.000	0.130	0.800	2.059	2.095-2.047
40.000	0.130	0.900	2.046	2.082-2.033
40.000	0.130	1.000	2.090	2.126-2.078
40.000	0.140	0.500	2.047	2.083-2.034

5 Conclusions

The advantage with this method is that once the results obtained were optimized to reduce the roughness and these values are sent to the machine, the result is a roughness very close to the limits shown in the table above, in the Table 5 shows the optimization values obtained for both neural networks. With these parameters, the next step is to bring the machine and carry out tests to verify the results generated by the network, although we can see that the results are very similar, so one can conclude that both are good, with the results of maximum sensitivity being obtained much faster.

Table 5. Predicted values with backpropagation and NNMS

Neural Network	cutting speed	feed rate	depth cut	roughness
Backpropagation	80	0.13	0.5	1.0584
NNMS	71	0.13	0.5	0.96

This shows that the neural network can help us predict the roughness, and this can be used as a complement when designing parts. It is very important to find the ideal conditions to permit machining optimization and reduce costs and the same time, which can be simple using an intelligent system.

6 Future Work

It is convenient to keep working on techniques to improve these methodologies (using evolution computation). Research is being done to integrate multi-objective algorithms considering the different variable combinations and constraints involved in the parts being machined. Analysis of different types of materials in different conditions within the process, such as super alloys, is needed because this type of material has poor machinability and the process is very expensive.

References

[1] Meziane, F., Vadera, S.: Intelligent systems in manufacturing: current developments and future Integrated manufacturing Systems, vol. 11, pp. 218–238 (2000)
[2] Morales, R., Vallejo, A., Avellan, J.: AI approaches for cutting tool diagnosis in machining processes. In: Proceedings of the 25th IASTED 978-0-88986-629-4, pp. 186–191 (2007)
[3] Pawadea, R.S., Suhas, S., Brahmankar, P.K.: Effect of machining parameters and cutting edge geometry on surface integrity of high-speed turned Inconel 718. International Journal of Machine Tools and Manufacture (2007), doi:10.1016/j.ijmachtools.2007.08.004
[4] He, W., Zhang, Y.F., Lee, K.S., Liu, T.I.: Development of a fuzzy-neuro system for parameter resetting injection molding. Transactions of the ASME 123 (February 2001)
[5] Rico, L., Díaz, J.: Surface roughness prediction at 1018 cold rolled steel using Response Surface Methodology and neural networks. Culcyt Research Year 2(10) (2005)
[6] Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn., pp. 111–114. Prentice Hall, Upper Saddle River (2003)
[7] Ramesh, S., Karunamoorthy, L., Ramakrishnan, R.: Modeling for prediction of surface roughness in machining of Ti64 alloy using response surface methodology. Journal of Materials Processing Technology (2007), doi:10.1016/j.jmatprotec.2007.11.031
[8] Che-Haron, C.H., Jawaid, A.: The effect of machining on surface integrity of titanium alloy Ti–6% Al–4% V. Journal of Materials Processing Technology 166, 188–192 (2005)
[9] Molinari, A., Musquar, C., Sutter, G.: Adiabatic shear banding in high speed machining of Ti–6Al–4V: experiments and modeling. International Journal of Plasticity 18, 443–459 (2002)
[10] Krain, H., Sharman, A., Ridgway, K.: Optimization of tool life and productivity when end milling Inconel 718 M. Journal of materials processing technology 189, 153–161 (2007)

[11] Kopac, J., Bahor, M., Sokovic, M.: Optimal machining parameters for achieving the desired surface roughness in fine turning of cold preformed steel workpieces. International Journal of Machine Tools and Manufacture, 42707–42716 (2002)
[12] Oktem, H., Erzurumlu, F.: Prediction of minimum surface roughness in end milling mold parts using neural network and genetic algorithm Materials and Design. Journal 27, 735–744 (2006)
[13] Egiazaryan, S.K., G.G.: Theory of functional systems in the scientific school of p.k. anokhin. Journal of the History of the Neurosciences 16(1-2), 194–205 (2007)
[14] Anokhin, P.: Biology and Neurophysiology of the Conditioned Reflex and Its Role in Adaptive Behavior. Pergamon, Oxford (1974)
[15] Anojin, P.K.: Psicología y la filosofía de la ciencia: Metodología del sistema funcional. editorial Trillas, México (1985)
[16] Red'ko, V.G., Prokhorov, D.V., Burtsev, M.S.: Theory of functional systems, adaptive critics and neural networks. In: Proceedings of IJCNN, pp. 1787–1792 (2004)
[17] Torres-Treviño, L.M.: Controladores dinámicos con la red neuronal de máxima sensibilidad. Master's thesis, Autonomous University of san Luis Potosi, San Luis Potosí, México (1998)
[18] Carlos González González y Ramon Zeleny, Metrología Dimensional. Mc Graw Hill
[19] Kim, B., Kim, S.: GA-optimized back propagation neural network with multi-parameterized gradients and applications to predicting plasma etch data. Chemometrics and Intelligent Laboratory Systems 79, 123–128 (2005)
[20] Basheer, I.A., Hajmeer, M.: Artificial neural networks: fundamentals, computing, design, and application. Journal of Microbiological Methods 43, 3–31 (2000)

Comparing Three Simulated Strategies for Cancer Monitoring with Nanorobots

C.A. Piña-García[1], E.J. Rechy-Ramírez[2], and V.A. García-Vega[1]

[1] Facultad de Física e Inteligencia Artificial, Departamento de Inteligencia Artificial, Universidad Veracruzana, Sebastián Camacho No 5, Xalapa, Ver., México, 91000
capiga21@gmail.com, angegarcia@uv.mx
[2] Laboratorio Nacional de Informática Avanzada, LANIA, Rébsamen 80, esq. Circuito Presidentes C.P. 91000 A.P. 696, Col. Centro. Xalapa, Veracruz, México
erechy@lania.mx

Abstract. The use of nanorobots in medical applications, specifically cancer treatment, is a serious alternative to prevent this disease. Locating chemical sources and tracking them over time, are tasks where nanorobotics is an ideal candidate to accomplish them. We present a multiagent simulation of three bio-inspired strategies to find targets in fluid environments; including diverse conditions for example: noisy sensors, interference between agents and obstacles generated by the environment itself. Besides, we present a comparative analysis among the three strategies. The results show that nanorobotics used in cancer therapy needs to explore an extensive range of blind searching techniques without communication.

1 Introduction

Nanomedicine is a prominent field of study which has many topics in its domain, one of these subjects is nanorobotics, particularly applied in cancer treatment [1,2,3,4,5]. This kind of research is a serious option to prevent cancer. Institutes as the National Cancer Institute are working with alternatives based on nanotechnology [8,9,10,11]. Nanorobotics architecture based on nanobioelectronics has been proposed for the gradual development and future use of nanorobots to combat cancer. A nanorobot team with embedded chemical biosensors, can be used to perform detection of tumor cells in early stages of development inside the body of the patient [1,2].

This paper presents three searching blind strategies with the goal of finding chemical sources (targets) in the bloodstream. This work is based on Cavalcanti's study and our previous research to suggest bio-inspired techniques for nanorobotics exploration [1,2,3,4,5,6,7]. It is necessary to find an efficient strategy for nanorobot exploration, to take into account the properties of nanorobots to navigate as bloodbome devices [4,12,13,14], if they are going to assist on vital aspects of cancer treatment.

The remainder of this paper is organized as follows. Section 2 discusses the synthetic design of the environment and nanorobots. Section 3 presents the description of the three bio-inspired strategies and how they were implemented.

A. Gelbukh and E.F. Morales (Eds.): MICAI 2008, LNAI 5317, pp. 1020–1030, 2008.

Section 4 discusses our results obtained in our simulator. Finally, section 5 concludes the paper.

2 Synthetic Design of the Environment and Nanorobots

One of the most important goals in cancer therapy, is to locate tumor cells in early stages of development to prevent their growth; hence, detecting chemical sources related with malignant tumors (targets for nanorobots) is an indispensable task to achieve [5,10,15,16]. Even more, nanorobotics architectures are correlated with advances in nanomaterials and manufacturing of nanotechnology; where the future use of these devices within the human body is imminent [5,16]. Therefore, it is necessary to experiment with already tested strategies (e.g. collective robotics). A higher performance with less amount of computing is our goal. Next, we explain how we model our virtual environment and the natural forces within.

2.1 Environment Properties

The aim of the environment design is to obtain a working micro-world used as a test bed for our examination, therefore it must correspond to a scalable segment from the complete searching area. Hogg [12] suggests a virtual microenvironment could be modeled considering a blood vessel section only. Parameters and values recommended by Hogg are listed in table 1.

Table 1. Parameters and Values Proposed by Tad Hogg

Parameter	Value
vessel and target	
vessel radius	$R = 5\mu m$
section length	$L = 100\mu m$
target length	$L_{target} = 30\mu m$
fluid	
fluid density	$\rho = 1g/cm^3$
fluid viscosity	$\eta = 10^{-2}g/$cm-s
average fluid velocity	$v_{avg} = 1000\mu m/s$
robots	
robot radius	$a = 1\mu m$

We use a bi-dimensional space, enough to obtain significant results [17,15]. Our micro-world has viscous properties where friction, adhesion and viscous forces are dominant and gravitational forces are not important [11,15,16,12]. Our workspace is a simple cut blood vessel, delimited by two "walls", upper and lower walls. The right and left sides of the blood vessel, have not boundaries like a torus shape. This allows us to keep a continuous flow across the vessel segment, simulating an endless section.

Another key point of our model is the dynamic behavior of the whole system. The model is generated by internal features or interactions among the elements of the world e.g. red blood cell motion, red blood cell collision, fluid motion and tumor growth. Red blood cell motion is governed by physics of fluid micro-environment [1,2,3,4]. Therefore, objects experience inertial forces from an entity with a characteristic length L; moving with velocity υ through a fluid viscosity η and a fluid density ρ, given by the Reynolds number:

$$Re = \frac{\rho \upsilon L}{\eta} \tag{1}$$

According with [17,12], using typical values for density and viscosity of blood plasma; a nanorobot has $Re \approx 10^{-3}$ with respect to the object velocity and due to the small area of the vessel occupied by the robots. We consider a constant velocity and ignore some stochastic factors for simplicity.

2.2 Nanorobot Properties

It is essential to make use of collective robotics techniques for the purpose to take advantage from already tested strategies [18,19,20,21,22]. We pretend to explore and experiment between three strategies with the aim to find a balance among performance and computing. One of the most crucial features of our methodology is the number of nanorobots. Different methods use a value of $N = \{20\}$ nanorobots to set up the simulator as a reasonable amount for a section length of $L = 100\mu m$ [2,12]; we use these metrics because we are pretending to work within a human body and optimization is a major priority. Therefore, we propose to use a lower value of N for our research purposes, $N = \{5\}$ for L.

Our nanorobots "scouts" are modeled as a homogeneous multi-agent system, they are able to search targets in a dynamic environment. Targets are non uniformly distributed over the segment $[-L, L]$; they are located in random positions along the upper and lower walls. There are three types of targets in our environment. Nanorobots can sense targets through their biochemical sensors, using receptors to detect specific chemicals [1,2,4,12]. Last cited papers operate in an "ideal" world, i.e. they do not consider noise levels in their proposed sensors. We design a kind of "decoys", to introduce noise disturbing the scout search.

Biochemical sensors are not useful to avoid obstacles (e.g. another agents). For this reason, we implement a potential field methodology to avoid other agents [15,23,24,25,26]. We use potential fields since they can be a function of the stimulus, regardless of distance. Moreover, we assume nanorobots do not have memory; hence the robot computes the effect of the potential field usually as a straight line, at every update, with no memory of where it was previously or where the robots have moved. Our design is based on a "repulsive" field with a linear drop off (see formula 2).

$$\overrightarrow{v}_{direction} = -180°$$

$$\overrightarrow{v}_{magnitude} = \begin{cases} \frac{(D-d)}{D} & \text{for } d \leq D \text{ and } D \neq 0 \\ 0 & \text{for } d > D \end{cases} \quad (2)$$

where D is the maximum range of the effect created by the potential field and; d is the real distance sensing by nanorobot. We use the Euclidean metric [17,25,26,27] to calculate distance in $\mathbb{R}^2$, see equation 3.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$

Then, if a robot r_1 is in radius D with respect to another robot r_2, and exists a minimum distance $min\{d\}$ between r_1 and r_2; our repulsive potential field triggers, and $\overrightarrow{v}$ takes values from the above formula.

Nanorobots have to move through the fluid, so they face significant drag [17,3,15,12,28,26]. Since we are modeling small objects at slow speeds, we use the equation for viscous resistance or linear drag to simulate the force drag $\mathbf{F}_d$ (see equation 4).

$$\mathbf{F}_d = -b\mathbf{v} \quad (4)$$

where b is a constant that depends on the properties of the fluid and the dimensions of the object; and $\mathbf{v}$ is the velocity of the object. Since we assume our agents as small spherical objects moving slowly through a viscous fluid, we use the Stokes derived expression for the drag constant b [17].

$$b = 6\pi\eta r \quad (5)$$

where r is the Stokes radius of the object, in this work robot radius ($1\mu m$); and η is the fluid viscosity $10^{-2} g/\text{cm-s}$.

Communication in our model is achieved by using implicit communication through the world (stigmergy) [18,19,20,21,22,25,27], robots sense the effects of the actions produced by teammates. We select stigmergy, because of its simplicity and its lack of dependence upon explicit communication channels and protocols. Figure 1 illustrates our environment and multi-agent system.

3 Description of the Three Strategies

In this section, three bio-inspired strategies used to find targets in fluid environments are discussed within a simulated application of a multiagent system. Our aim is the use of this approach in medical applications, specifically to the cancer treatment.

3.1 Wiggle Strategy

This approach is based on natural/artificial selection model, showing the interrelationship between natural selection forces in predator and prey that results

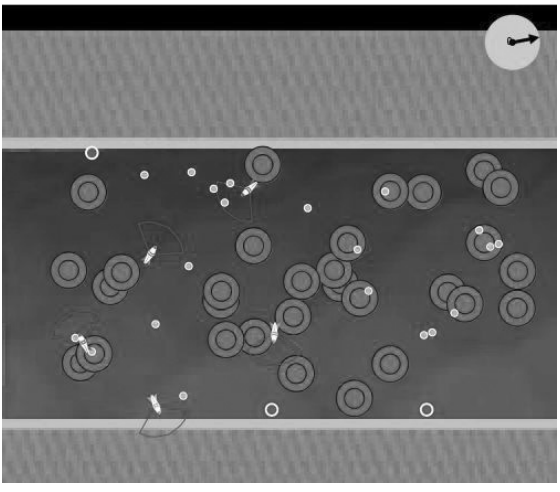


Fig. 1. Environment design with the nanorobots within it, targets are symbolized with circles in the upper wall and the lower wall. Red blood cells and other objects produce noise in the micro-world.

from coevolution [23,25,29]. The model demonstrates many different series of reciprocal changes between a predator and prey; which each acts as an agent of natural selection for each other, particularly related to the successive adaptation of the predator for pursuing its prey [19,30,31,25,29,27]. We make use of this methodology, to present a search task inside a blood vessel section previously mentioned in section 2.1. Furthermore, we adapt the Novak model [29] to a non evolutionary perspective with the meaning of keeping a constant bearing in each nanorobot. Wiggle strategy is a simple procedure, where the agents are sensing at the same time they are doing the "wiggle". In this case, we implement the "wiggle" with random movement interchange. This swapping is achieved by alternating turn right or turn left in a closed interval $\{\theta : 0° \leq \theta \leq 30°\}$, where θ is the angle to turn.

3.2 Saltatory Strategy

"Saltatory" strategy presents an intermittent behavior, commonly observed in species as large fishes, rattlesnakes, birds, lizards, etc. This intermittent behavior is used when the targets are difficult to detect, and sparsely distributed; that is, when predator has no information about the prey location and the preys are immobile and randomly distributed with uniform density [32]. This strategy consists of two phases described below:

1. **search phase:** each nanorobot explores its immediate vicinity using its biochemical sensors,
2. **motion phase:** according to [32], the searcher moves rapidly and is unable to detect a target. However, in our exploration agents are situated in a viscous fluid that presents an important drag (see section 2.2); and preys/targets are not arbitrarily disseminated (i.e. non uniform target density) as we pointed out in section 2.2; hence, we adjust the strategy to keep using the sensor during the second phase.

Phases switching is based on time intervals, namely ticks according to the tool we use to model our environment (Netlogo [33]). These intervals are defined as follows: $\{T : 0 \leq T \leq 600\}$ correspond to the most general interval, $\{\tau : 0 \leq \tau \leq 200\}$ correspond to the partition of T, one for each nanorobot. Subsequently, we divide T according to n the number of nanorobots, as follows: $\forall nanorobot_i := \tau$ where $i \in \{0, 1, 2, 3, 4\}$ then

$$T = \bigcup_{i \cap i+1}^{n=4} \tag{6}$$

3.3 Wandering Strategy

The wander behavior controls an agent moving aimlessly about [17,23]. We perturbed the wander direction by a random amount. This makes the nanorobot moves forward smoothly, but the rotation of the nanorobot is erratic, appearing to switch from side to side as it moves [23,27]. The agent tries to face the target in each step, using sensors to align to the target. Then, the nanorobot applies full acceleration in the direction of its current orientation. This wander behavior biases the agent to turn (in either direction). The agents start at time $t = 0$ at random position (x, y) and execute a random walk according to the following rules:

1. each nanorobot turns to the right or to the left when it sense another agent or a wall; this is possible because of the repulsive potential field. Our nanorobots are moving at velocity $\pm v_{x,y}$ in τ ticks; so we have a distance $\delta = \pm v_{x,y}\tau$. For simplicity, we treat $v_{x,y}$ as a constant,
2. the probability of going to the right or to the left at each step is biased by the interaction with another nanorobots, and the influence exerted by the "decoys" since; if nanorobots follow a decoy they can change their route,
3. each nanorobot moves independently of all other nanorobots. The nanorobots do not interact with another one.

Figure 2 shows the plots obtained from our simulator for the three strategies. Based on the terrain coverage wandering strategy is more efficient than wiggle and saltatory strategies, wandering strategy aids to explore more terrain. Nevertheless, we corroborate this assumption in section 4.

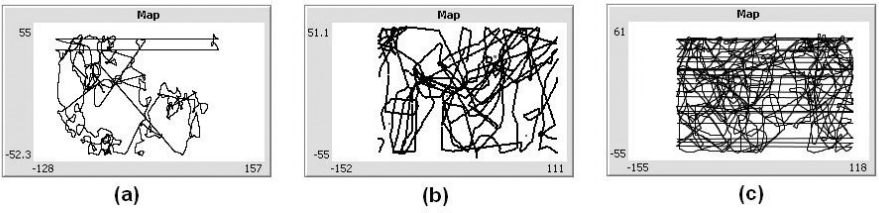


Fig. 2. An x, y plot of a two-dimensional random walk of each strategy, (a) belongs to wiggle strategy, (b) is the saltatory strategy and (c) is the wandering strategy

4 Results

In this section, we describe the tests on our tool and results. We did twenty five tests for each strategy in two laptops, W (Windows XP, Intel Celeron, 1.70GHz) and M (MAC OS X 10.4.11 Intel Core 2 Duo, 2GHz). The tests were executed with the following parameters: the initial speed of nanorobots was 0.50, the number of red blood cells was 30, the initial nanorobot sensor was 4.0 and the noise-saturation (decoys) was 20 (see figure 3). The end of test came when the five nanorobots found the three targets; we took the time required to accomplish this.

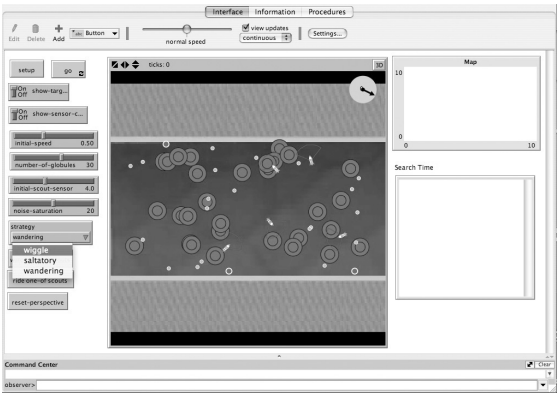


Fig. 3. Interface of the simulator

Figure 4 shows the time used by nanorobots to find the targets, for each of the three strategies in the mac computer. Time is given in minutes. The wiggle strategy is the worst strategy, in the 25 tests its best time was 0.21 minutes; while the best times for the saltatory and wandering strategies, were 0.09 and 0.06 minutes, respectively. The worst time for the wiggle strategy in the 25 tests

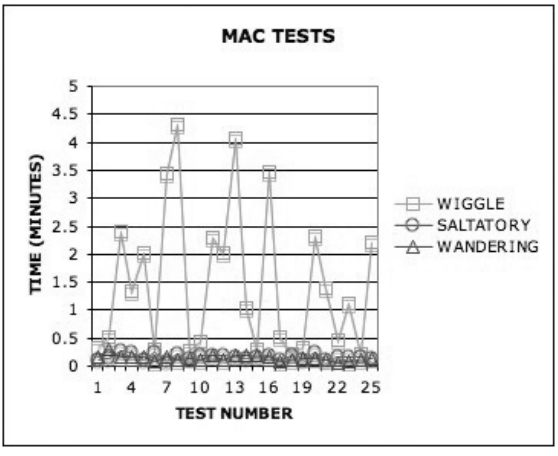


Fig. 4. The twenty five tests executed in mac computer

Table 2. Average times in minutes for the three strategies in mac and windows computers

Strategy	*Mac*	*Windows*
Wiggle	1.49	2.50
Saltatory	0.17	0.59
Wandering	0.14	0.33

was 4.32; while the worst times for the saltatory and wandering strategies were 0.28 and 0.29 minutes, respectively.

Figure 5 shows the time in minutes, used by nanorobots to find the targets for the three strategies; but now for the windows computer. This plot also shows wiggle strategy as the worst strategy for exploration. In the 25 tests the best time for this strategy was 0.45 minutes; while the best time of saltatory and wandering strategies was the same, 0.15 minutes. The worst time for the wiggle strategy, was 8.03; while the worst times for the saltatory and wandering strategies were 1.24 and 0.59 minutes, respectively.

Table 2 presents the average times for the strategies. In the mac computer, we found that the average times for the wiggle, saltatory and wandering strategies were 1.49, 0.17 and 0.14 minutes, respectively. While in the windows computer, the average times for the wiggle, saltatory and wandering strategies were 2.50, 0.59 and 0.33 minutes, respectively.

In conclusion, we found that the strategy that took more time in detecting the three targets in both equipments, was the wiggle strategy. Nevertheless, the

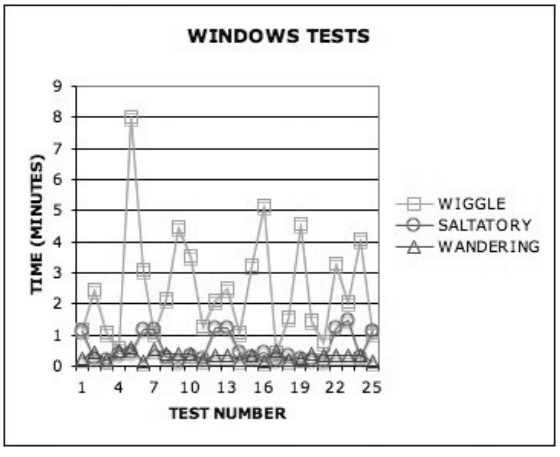


Fig. 5. The twenty five tests executed in windows computer

strategy that took less time in finding the targets in both computers was the wandering strategy. The results for the winner and worst strategies are kept across different equipment, they did not give substantial differences.

5 Conclusions and Future Work

In our research, nanorobots using bio-inspired search techniques were programmed to find among different strategies the most effective one. With our study have initiated a line of research for finding tumor cells in early stages of development in a small piece of a blood vessel; where we use a minimum of robots for maximum performance.

The results obtained in the simulation; indicated that simple search strategies, can be incorporated to a nanorobot team with the aim to find targets in a dynamic environment. The trials revealed how each technique responds to the micro-world stimuli; and the most efficient strategy, in this case was wandering strategy. The tests confirm that wandering strategy is a practical method to monitor tumor cells in early stages of development (targets). This strategy makes possible a better performance for nanorobots to perceive possible targets.

Research of nanorobot search methods, can result in important improvement of architecture for nanorobot application in cancer therapy, and advances in nanorobotics simulation.

Although, we just experiment with three strategies; we are now in a position to gain fundamental insight into how nanorobots could search inside our body. We are developing a new strategy based on an experimental niche, that can present a better performance locating targets. Furthermore, we are developing a new class of nanorobots to conform a heterogeneous team, that can be used to deliver medicine in the affected region.

References

1. Cavalcanti, A., Shirinzadeh, B., Zhang, M., Kretly, L.C.: Nanorobot Hardware Architecture for Medical Defense. Sensors (2008) ISSN 1424-8220
2. Cavalcanti, A., Shirinzadeh, B., Hogg, T., Smith, J.A.: Hardware Architecture for Nanorobot Application in Cancer Therapy. In: IEEE-RAS ICAR Intl. Conf. on Advanced Robotics, Jeju, Korea, pp. 200–205 (August 2007)
3. Cavalcanti, A., Shirinzadeh, B., Fukuda, T., Ikeda, S.: Hardware Architecture for Nanorobot Application in Cerebral Aneurysm. In: IEEE - Nano 2007 Intl. Conf. on Nanotechnology
4. Cavalcanti, A., Shirinzadeh, B., Freitas Jr., R.A., Kretly, L.C.: Medical Nanorobot Architecture Based on Nanobioelectronics. Recent Patents on Nanotechnology 1, 1–10 (2007)
5. Cavalcanti, A., Shirinzadeh, B., Murphy, D., Smith, J.A.: Nanorobots for Laparoscopic Cancer Surgery. In: IEEE ICIS 2007 International Conference on Computer and Information Science
6. Piña-García, Adolfo, C., Garcia-Vega, V.A.: A Hybrid Methodology for Robotic Architectures with a Cellular Approach. E-Learning in Industrial Electronics. In: 2006 1ST IEEE International Conference on (ICELIE), pp. 156–160 (December 2006)
7. Piña-García, C.A., Rechy-Ramírez, E.J., García-Vega, V.A.: Using an Alternative Model in a Complex Environment for Nanorobotics Navigation. In: 16th International Conference on Computing (CIC) (November 2007)
8. Cavalcanti, A., Hogg, T., Shirinzadeh, B., Liaw, H.C.: Nanorobot Communication Techniques: A Comprehensive Tutorial. In: IEEE ICARCV 2006 International Conference on Control, Automation, Robotics and Vision (2006)
9. Cavalcanti, A., Rosen, L., Shirinzadeh, B., Rosenfeld, M.: Nanorobot for Treatment of Patients with Artery Occlusion. In: Proceedings of Virtual Concept 2006, Cancun, Mexico (2006)
10. Cavalcanti, A.: Robots in Surgery. In: Euro Nano Forum 2005, Nanotechnology and the Health of the EU Citizen in 2020 (2005)
11. Cavalcanti, A., Freitas Jr., R.A.: Nanosystem Design with Dynamic Collision Detection for Autonomous Nanorobot Motion Control using Neural Networks. Computer Graphics and Geometry Journal 5(1), 50–74 (2003)
12. Hogg, T.: Coordinating Microscopic Robots in Viscous Fluids. HP Labs Palo Alto, CA. October 9 (2006)
13. Pecina-Slaus, N. (checar) Tumor suppressor gene E-cadherin and its role in normal and malignant cells. Cancer Cell International 3, 17 (2003)
14. Ummat, A., Sharma, G., Mavroidis, C., Dubey, A.: Bio-Nanorobotics: State of the Art and Future Challenges. Northeastern University (2005)
15. Dhariwal, A., Sukhatme, G.S., Requicha, A.A.G.: Bacterium-inspired Robots for Environmental Monitoring. In: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA (April 2004)
16. Drexler, E.K.: Engines of Creation. Anchor Press (1986)
17. Berg, H.C.: Random walks in biology. Princeton University Press, Princeton (1993)
18. Arkin, R.C.: Cooperation without communication: Multiagent schemes-based robot navigation. Journal of Robotic Systems 9(3), 351–364
19. Cao, Y.U., Fukunaga, A.S., Kahng, A.B.: Cooperative Mobile Robotics: Antecedents and Directions. Autonomous Robots 4, 123 (1997)

20. Defago, X.: Distributed computing on the move: From mobile computing to cooperative robotics and nanorobotics. In: Proc. 1st ACM Int'l Workshop on Principles of Mobile Computing (POMC 2001), Newport, RI, USA, pp. 49–55 (August 2001)
21. Dorigo, M.: Swarm-Bot: An Experiment in Swarm robotics. IEEE, Los Alamitos (2005)
22. Kube, C.R., Zhang, H.: Collective robotics: from social insects to robots. Adaptive Behavior 2, 189218 (1994)
23. Millington, I.: Artificial Intelligence for games, 1st edn. Morgan Kaufmann, San Francisco (2006)
24. Miller, M.B., Bassler, B.L.: Quorum sensing in bacteria. Annu. Rev. Microbiol. 55, 16599 (2001)
25. Murphy, R.R.: Introduction to AI robotics. The MIT Press, Cambridge (2000)
26. Palmer, G.: Physics for Game Programmers. Apress (2005) ISBN10: 1-59059-472-X
27. Siciliano, B., Khatib, O.: Handbook of robotics. Springer, Heidelberg (2008)
28. Listak, M., Martin, G., Pugal, D., Aabloo, A., Kruusmaa, M.: Design of a Semiautonomous Biomimetic Underwater Vehicle for Environmental Monitoring. In: 6th IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA 2005), Espoo, Finland, 27.06.-30.06.2005, pp. 9–14. IEEE, New York (2005)
29. Novak, M., Wilensky, U.: NetLogo Bug Hunt Coevolution model. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL, http://ccl.northwestern.edu/netlogo/models/BugHuntCoevolution
30. Ghose, K., Horiuchi, T.K., Krishnaprasad, P.S., Mos, C.F.: Echolocating Bats Use a Nearly Time-Optimal Strategy to Intercept Prey. PLoS Biology 4(5), 108 (2006)
31. Ghose, K., Horiuchi, T.K., Krishnaprasad, P.S., Moss, C.F.: The echolocating bats pursuit strategy is optimal for unpredictably maneuvering prey. NACS 79.1
32. Bénichou, O., Coppey, M., Moreau, M., Suet, P.-H., Voituriez, R.: Optimal Search Strategies for Hidden Targets. The American Physical Society (2005)
33. Wilensky, U.: Netlogo, center for connected learning and computer-based modeling (1999), http://ccl.northwestern.edu/netlogo

Author Index