Angel Kuri-Morales
Guillermo R. Simari (Eds.)

LNAI 6433

Advances in Artificial Intelligence – IBERAMIA 2010

12th Ibero-American Conference on AI
Bahía Blanca, Argentina, November 2010
Proceedings

Springer

Lecture Notes in Artificial Intelligence 6433

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

Subseries of Lecture Notes in Computer Science

Angel Kuri-Morales Guillermo R. Simari (Eds.)

Advances in Artificial Intelligence – IBERAMIA 2010

12th Ibero-American Conference on AI
Bahía Blanca, Argentina, November 1-5, 2010
Proceedings

Springer

Volume Editors

Angel Kuri-Morales
Instituto Tecnológico Autónomo de México
Departamento Académico de Computación
Río Hondo No. 1, México 01000 D.F., México
E-mail: akuri@itam.mx

Guillermo R. Simari
Universidad Nacional del Sur
Department of Computer Science and Engineering
Alem 1253, 8000 Bahía Blanca, Argentina
E-mail: grs@cs.uns.edu.ar

Library of Congress Control Number: 2010938220

CR Subject Classification (1998): I.2.6-7, I.2.9, H.3-5, I.4, I.2.3, I.2.11

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-16951-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-16951-9 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

IBERAMIA is the international conference series of the Ibero-American artificial intelligence community, which has met every two years since the 1988 meeting in Barcelona. The conference is supported by the main Ibero-American societies of AI and provides researchers from Latin America, Portugal, and Spain the opportunity to meet AI researchers from all over the world. Since its inception IBERAMIA has been a widely recognized international conference. Its papers have been written, presented, and published in English. Papers of the main track have been published by Springer in the LNCS/LNAI series. This volume contains the papers accepted for presentation at IBERAMIA 2010, held in Bahía Blanca, Argentina, in November 2010. A total of 178 papers were submitted for the main track, and 61 papers were accepted for publication and oral presentation. Each paper was reviewed by five members of the Program Committee (PC), coordinated by a PC Chair. The PC consisted of 110 researchers from 74 universities and research centers from 17 different countries. The statistics of the PC members are shown in the following table; the full list of PC members and reviewers can be found on the pages that follow.

Statistics by Country (PC Members)

Country	Members
Argentina	8
Australia	2
Austria	1
Brazil	14
Chile	2
Colombia	1
Czech Republic	1
France	1
Germany	2
Japan	1
Mexico	34
Netherlands	1
Poland	2
Portugal	10
Spain	18
Sweden	1
UK	4
USA	1
Venezuela	1

The authors of the submitted papers represent members of 24 countries, 9 from Ibero-America and 15 from other regions of the world, attesting to the truly international nature of the conference. The topics cover 23 main areas: commonsense reasoning; cognitive modeling and human interaction; uncertainty and fuzzy systems; game playing and interactive entertainment; probabilistic reasoning; constraint satisfaction; distributed AI; argumentation; genetic algorithms; model-based systems; search; AI in education; planning and scheduling; evolutionary computation and artificial life; semantic web; knowledge acquisition and ontologies; robotics, vision, and pattern recognition; natural language processing; multiagent systems; information integration and extraction; neural networks; knowledge representation and reasoning; and machine learning and data mining.

Statistics by Country (Papers)

Country	Authors	Submitted	Accepted
Argentina	64	26.83	6.00
Brazil	134	58.13	15.67
Chile	7	3.00	1.00
Colombia	3	2.00	-
Cuba	4	4.00	-
Czech Republic	1	1.00	-
France	3	1.50	1.50
South Korea	1	1.00	0.00
Mexico	118	51.55	14.17
Pakistan	1	1.00	-
Portugal	20	8.33	1.33
Spain	48	15.37	3.50
Tunisia	2	1.00	-
UK	1	0.33	0.33
USA	4	0.95	0.50

The program for IBERAMIA 2010 also included three invited speakers: Helder Coelho (On Selecting and Guiding AI PhD Students), Jaime Carbonell (Active and Proactive Machine Learning), and Carles Sierra (Information-Based Trust and Reputation). Dr. Coelho is full professor of the University of Lisbon (UL), in the Department of Informatics of the Faculty of Sciences. Dr. Carbonell is Director of the Language Technologies Institute and Allen Newell Professor of Computer Science at Carnegie Mellon University. Dr. Carles Sierra is professor at the Artificial Intelligence Research Institute (IIIA) of the Spanish Research Council (CSIC) in Bellaterra, Barcelona.

Papers where students were involved were carefully analyzed by a selection committee. A Best Student Paper prize was awarded to Eunice López Camacho from the Tecnológico de Monterrey, Mexico, for the paper *Defining a Problem-State Representation with Data Mining within a Hyper-heuristic Model which Solves 2D Irregular Bin Packing Problems*. Special Student Paper

Mentions were awarded to Andrea Cohen from the Universidad Nacional del Sur, Argentina, for the paper *Extending DeLP with Attack and Support for Defeasible Rules*; Augusto Baffa from the Universidade Federal do Estado do Rio de Janeiro, Brazil, for the paper *Planning under the Uncertainty of the Technical Analysis of Stock Markets*, and Wildener Monteiro Rodovalho from the Universidade Federal de Goiás, Brazil, for the paper *Studying the Emergence of Money by Means of Swarm Multi-Agent Simulation*.

We want to thank all those involved in the organization of this conference. In the first place, these are the authors of the papers constituting this book: it is the excellence of their research work that gives value to the book and sense to the work of all of the other people involved. We thank the members of the Program Committee and additional reviewers for their great and very professional work on reviewing and selecting the papers for the conferences. Our very special thanks are due to the local organizing committe, the members of the Department of Science and Engineering of Computation of the Universidad Nacional del Sur. We wish to express our deepest gratitude to the students of the Doctoral Program of Ciencia e Ingeniería de la Computación of the Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) at UNAM. Special thanks are also due to Dr. José Galaviz Casas, for his participation in the committee to select the best students paper. An activity which was re-instated for IBERAMIA 2010. Last but not least, we wish to express our gratitude to Edwin Aldana-Bobadilla for his invaluable help throughout the process of arbitration and edition of these proceedings.

November 2010 Angel Kuri-Morales

Organization

IBERAMIA 2010 was organized by the Department of Computer Science and Engineering, Universidad Nacional del Sur (DCSE-UNS), Argentina. The conference was sponsored by the main Ibero-American artificial intelligence and computer science societies.

Steering Committee

Federico Barber	AEPIA, Spain
Helder Coelho	Universidade de Lisboa, Portugal
Francisco Garijo	Telefónica I+D, Spain
Christian Lemaître	LANIA, Mexico
Arlindo Oliveira	APIA, Portugal
Carlos Alberto Reyes García	SMIA, Mexico
Jaime Sichman	SBC, Brazil

Program Committee

Program Chair: Angel Kuri	ITAM, Mexico
Workshop Chair: Guillermo R. Simari	Universidad Nacional del Sur, Argentina
Student Chair: José Galaviz	UNAM, Mexico

Technical Committee

Abel Ortega Pacheco
Daniel Trejo Baños
Edwin Aldana Bobadilla
Héctor Félix Vidrios
Marco Antonio Becerra Pedraza

Local Organization Committee

Marcelo Alejandro Falappa
Alejandro Javier García
Carlos Iván Chesñevar
Diego César Martínez
Marcela Capobianco
Sergio Alejandro Gómez
Ana Gabriela Maguitman
María Paula González

Sponsoring Institutions

DCIC - Departamento de Ciencias e Ingeniería de la Computación
UNS - Universidad Nacional del Sur
AIJ - Artificial Intelligence Journal

Special thanks are due to the members of the Department of Science and Engineering of Computation of UNS (Departamento de Ciencias e Ingeniería de la Computación de la UNS).

Reviewers

Abdennou El Rhalibi	Liverpool John Moores University
Abel Ortega Pacheco	Universidad Nacional Autónoma de México
Alejandro Peña Ayala	Instituto Politécnico Nacional
Ana Casali	Universidad Nacional de Rosario
Ana Laureano-Cruces	Universidad Nacional Autónoma de México
Analia Amandi	UNICEN University
Angel Kuri-Morales	Instituto Tecnológico Autónomo de México
Antonio Bahamonde	Universidad de Oviedo
Bedrich Benes	Purdue University
Broderick Crawford	Pontificia Universidad Católica de Valparaíso
Carlos A. Reyes Garcia	Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE)
Carmelo Del Valle	Universidad de Sevilla
Cesar Analide	Universidade do Minho
Christian Lemaître León	Universidad Nacional Autónoma de México
Claudia Zepeda Cortes	Benemérita Universidad Autónoma de Puebla
Constantin Orasan	University of Wolverhampton
Cristian Mateos Diaz	UNICEN University
Daniel Borrajo	Universidad Carlos III de Madrid
Daniel Trejo Baños	Universidad Nacional Autónoma de México
Daniela Godoy	UNICEN University
David Eduardo Pinto A.	Benemérita Universidad Autónoma de Puebla
Denis Omar Verdugo P.	Universidad Nacional Autónoma de México
Diego Martinez	Universidad Nacional del Sur
Dieter Hutter	German Research Center for Artificial Intelligence
Edgard Benítez-Guerrero	Centro de Investigación y Estudios Avanzados IPN
Eduardo Morales	Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE)

Efren Mezura-Montes	Laboratorio Nacional de Informática Avanzada (Lania)
Emmanuel Hegmann G.	Universidad Nacional Autónoma de México
Enrique Jaimes Islas	Universidad Nacional Autónoma de México
Eric Monfroy	Université de Nantes
Estevam Rafael Hruschka	Universidade Federal de São Carlos
Esther Colombini	Instituto Tecnológico de Aeronáutica
Fabio A. Gonzalez	Universidad Nacional de Colombia
Fabio Okuyama	Universidade Federal do Rio Grande do Sul
Federico Barber	Universidad Politécnica de Valencia
Fernando Morales	Instituto Tecnológico Autónomo de México
Fernando Santos O.	Universidade de São Paulo
Fernando Zacarias-Flores	Benemérita Universidad Autónoma de Puebla
Francisco Reinaldo	Universidade do Porto
Francisco Ribadas-Pena	Universidade de Vigo
Gabriela Henning	Universidad Nacional del Litoral
George D. Cavalcanti	CIn-UFPE
Gerald Steinbauer	Graz University of Technology
Grigori Sidorov	Centro de Investigación en Computación IPN
Guillermo Morales-Luna	Centro de Investigación y Estudios Avanzados IPN
Guillermo R. Simari	Universidad Nacional del Sur
Gustavo Arroyo-Figueroa	Instituto de Investigaciones Eléctricas
Héctor Félix Vidrios	Universidad Nacional Autónoma de México
Héctor Hugo Avilés A.	Universidad Nacional Autónoma de México
Helder Coelho	Universidade de Lisboa
Hiram Calvo	Centro de Investigación en Computación IPN
Humberto Sossa A.	Centro de Investigación en Computación IPN
Ildar Batyrshin	Mexican Petroleum Institute (IMP)
Jacek Malec	Lund University
Jaime Sichman	Escola Politécnica da Universidade de São Paulo
Javier García García	Universidad Nacional Autónoma de México
João Balsa	Universidade de Lisboa
Jomi Fred Hubner	Universidade Regional de Blumenau

José Galaviz Casas	Universidad Nacional Autónoma de México
Juan Alvarado	Universidad Nacional Autónoma de México
Juan Arturo Nolazco F.	Instituto Tecnológico de Estudios Superiores de Monterrey
Juan Carlos Nieves	Universitat Politècnica de Catalunya
Juan Manuel Corchado R.	Universidad de Salamanca
Juan Pavón	Universidad Complutense de Madrid
Lawrence Mandow	Universidad de Malaga
Luis Angel Contreras T.	Universidad Nacional Autónoma de México
Luis Antunes	Universidade de Lisboa
Luís Correia	Universidade de Lisboa
M. Ángel Salido G.	Universidad Politécnica de Valencia Spain
Manuel Mejia-Lavalle	Instituto de Investigaciones Eléctricas
Marcelo Errecalde	Universidad Nacional de San Luis
Márcio Porto Basgalupp	Escola Politécnica da Universidade de São Paulo
Marco Antonio Becerra P.	Universidad Nacional Autónoma de México
Maria Carolina Monard	Escola Politécnica da Universidade de São Paulo
María José Castro Bleda	Universidad Politécnica de Valencia
Mariá Nascimento	Escola Politécnica da Universidade de São Paulo
Mauricio Javier Osorio G.	Universidad de las Américas Puebla
Michael Maher	University of New South Wales
Miguel Toro	Universidad de Sevilla
Nareli Cruz Cortés	Instituto Politécnico Nacional
Nils Bulling	Clausthal University of Technology
Nuno David	Instituto Universitário de Lisboa
Oliver Boissier	Ecole des mines de Saint-Etienne
Oscar Herrera Alcántara	UAM Azcapotzalco
Osvaldo Cairó	Instituto Tecnológico Autónomo de México
Oswaldo Teran	Universidad de Los Andes
Paloma Moreda Pozo	University of Alicante
Paulo Quaresma	Universidade de Évora
Paulo Trigo	Instituto Superior de Engenharia de Lisboa
Pedro Barahona	Universidade Nova de Lisboa
Peter Novak	Czech Technical University in Prague
Rafael Murrieta	CIMAT

Ramon Brena	Instituto Tecnológico de Estudios Superiores de Monterrey
Ramon Zatarain	Instituto Tecnológico de Culiacán
Reinaldo Bianchi	Centro Universitário da FEI
Renata Wassermann	Universidade de São Paulo
Roque Marin Morales	Universidad de Murcia
Rosa Viccari	Universidade Federal do Rio Grande do Sul
Roseli Francelin Romero	Universidade de São Paulo
Ryszard Klempous	Wrocław University of Technology
Salvador Abreu	Universidade de Évora
Sascha Ossowski	Universidad Rey Juan Carlos
Serafin Moral	Universidad de Granada
Simon Colton	Department of Computing at Imperial College London
Slawomir Nowaczyk	Lund University
Sofia Galicia Haro	Universidad Nacional Autónoma de México
Toby Walsh	ICTA and UNSW Australia
Vicente J. Botti Navarro	Universidad Politécnica de Valencia
Victor Manuel Darriba Bilbao	Escuela Superior de Ingenireía e Informática
Virginia Dignum	Delft University of Technology
Wamberto Vasconcelos	University of Aberdeen

Additional Reviewers

Aciar, Silvana	Cintra, Marcos
Aldana-Bobadilla, Edwin	Crasso, Marco
Alvarez-Napagao, Sergio	Cruz Roa, Angel Alfonso
Amaral, Laurence	Eyharabide, Victoria
Andrade Filho, José Augusto	Fernandes, Leandro
Antiqueira, Lucas	Fernández, Fernando
Armentano, Marcelo	Garcia Gutierrez, Jorge
Armentano, Marcelo Gabriel	Garcia-Olaya, Angel
Barros, Rodrigo	Guelpeli, Marcus
Batista, Gustavo	Guil-Reyes, Francisco
Caicedo, Juan C.	Gómez-Sebastiá, Ignasi
Camargo-Brunetto, Maria Angelica	Hermoso, Ramon
Campos Martinez, Manuel	Hernandez-Orallo, Jose
Campos, Francisco	Kolesnikova, Olga
Carballido-Carranza, José Luis	Ledeneva, Yulia
Castro, Paulo Andre	Lima, Allan
Centeno, Roberto	Lopes, Nuno
Chaczko, Zenon	Lopez, Rigoberto

Luaces, Oscar
Maciejewski, Henryk
Manso, Antonio
Marcellino, Fernando J.M.
Mateos, Daniel
Medina, Maria Auxilio
Moniz, Luis
Monteserin, Ariel
Nardin, Luis Gustavo
Nascimento, Mariá
Olmos, Ivan
Palma, Jose
Pedro, Vasco
Pereira, Cesar Lima
Peres, Andre
Pessin, Gustavo
Peña, Luis
Pinilla, Vanesa
Plüs, Brian
Pontes, Beatriz
Prati, Ronaldo
Ramos da Silva, Renato
Reyes-Galaviz, Orion Fausto
Rodriguez, Juan Manuel
Ruiz, Ubaldo
Santana, Pedro
Shahamatnia, Ehsan
Szlachcic, Ewa
Troyano, José A.
Villanueva, Alicia
Yaich, Reda
dos Santos, Edimilson
van der Zwaan, Janneke

Table of Contents

Artificial Intelligence in Education

Building and Assessing Intelligent Tutoring Systems with an e-Learning 2.0 Authoring System 1
M.L. Barrón-Estrada, Ramón Zatarain-Cabada, Rosalío Zatarain-Cabada, Hector Barbosa-León, and Carlos A. Reyes-García

Cognitive Modeling and Human Reasoning

Robotic Orientation towards Speaker for Human-Robot Interaction 10
Caleb Rascón, Héctor Avilés, and Luis A. Pineda

Dialogue Model Specification and Interpretation for Intelligent Multimodal HCI 20
Luis A. Pineda, Ivan V. Meza, and Lisset Salinas

Constraint Satisfaction

Optimization of Investment Options Using SQL 30
Jose Torres-Jimenez, Himer Avila-George, Nelson Rangel-Valdez, and Jorge Martinez-Pena

A Computational Method for Defeasible Argumentation Based on a Recursive Warrant Semantics 40
Teresa Alsinet, Ramón Béjar, and Lluís Godo

Evolutionary Computation

Improving Efficiency of a Genetic Algorithm Applied to Multi-robot Tactic Operation 50
Gustavo Pessin, Fernando S. Osório, Denis F. Wolf, and Christiane R.S. Brasil

VarMOPSO: Multi-Objective Particle Swarm Optimization with Variable Population Size 60
Javier López, Laura Lanzarini, and Armando De Giusti

A Multi-Objective Genetic Algorithm with Path Relinking for the p-Median Problem 70
José E.C. Arroyo, Paula M. dos Santos, Michele S. Soares, and André G. Santos

Information, Integration and Extraction

Using Visual Metrics to Selecting ICA Basis for Image Compression: A Comparative Study 80
Patricia R. Oliveira, Henrique Morimitsu, and Esteban F. Tuesta

Knowledge Acquisition and Ontologies

Extending DeLP with Attack and Support for Defeasible Rules 90
Andrea Cohen, Alejandro J. Garcia, and Guillermo R. Simari

An Argumentation Machinery to Reason over Inconsistent Ontologies ... 100
Martín O. Moguillansky, Renata Wassermann, and Marcelo A. Falappa

Knowledge Representation and Reasoning

Planning under the Uncertainty of the Technical Analysis of Stock Markets 110
Augusto C.E. Baffa and Angelo E.M. Ciarlini

From Inconsistency to Consistency: Knowledge Base Revision by Tableaux Opening 120
Camilla Schwind

Nondeterministic Planning for Generating Interactive Plots 133
Fabio A. Guilherme da Silva, Angelo E.M. Ciarlini, and Sean W.M. Siqueira

Diagnosis of Medical Images Using an Expert System 144
Itzel Abundez Barrera, Eréndira Rendón Lara, Citlalih Gutiérrez Estrada, and Sergio Díaz Zagal

Revision of CTL Models 153
Paulo T. Guerra and Renata Wassermann

Applying H^m Heuristics in Petri Nets Reachability Problem 163
Rene Kultz, Luis Allan Künzle, and Fabiano Silva

Machine Learning and Data Mining

Sampling for Information and Structure Preservation When Mining Large Data Bases 174
Angel Kuri-Morales and Alexis Lozano

Improved Graph-Based Metrics for Clustering High-Dimensional Datasets 184
Ariel E. Bayá and Pablo M. Granitto

Empirical Evaluation of Ranking Prediction Methods for Gene Expression Data Classification 194
Bruno Feres de Souza, André C.P.L.F. de Carvalho, and Carlos Soares

Defining a Problem-State Representation with Data Mining within a Hyper-heuristic Model Which Solves 2D Irregular Bin Packing Problems 204
Eunice López-Camacho, Hugo Terashima-Marín, and Peter Ross

Digital Web Library of a Website with Document Clustering 214
Isabel Mahecha-Nieto and Elizabeth León

Learning to Discover Faulty Spots in cDNA Microarrays 224
Mónica G. Larese, Pablo M. Granitto, and Juan C. Gómez

A First Machine Learning Approach to Pronominal Anaphora Resolution in Basque 234
O. Arregi, K. Ceberio, A. Díaz de Illarraza, I. Goenaga, B. Sierra, and A. Zelaia

Multiagent Systems

A Holonic Multi-agent Model for Oil Industry Supply Chain Management 244
Fernando J.M. Marcellino and Jaime S. Sichman

Moral Minds as Multiple-Layer Organizations 254
Helder Coelho, António Carlos da Rocha Costa, and Paulo Trigo

Making Electronic Contracting Operational and Trustworthy 264
Joana Urbano, Henrique Lopes Cardoso, and Eugénio Oliveira

Normative Reasoning with an Adaptive Self-interested Agent Model Based on Markov Decision Processes 274
Moser Silva Fagundes, Holger Billhardt, and Sascha Ossowski

Query-Based Argumentation in Agent Programming 284
Sebastian Gottifredi, Alejandro J. Garcia, and Guillermo R. Simari

Studying the Emergence of Money by Means of Swarm Multi-agent Simulation 296
Wildener M. Rodovalho, Cássio Dener N. Vinhal, and Gelson da Cruz Júnior

Natural Language Processing

Revisiting the Readability Assessment of Texts in Portuguese 306
Carolina Scarton, Caroline Gasperin, and Sandra Aluisio

Improved Text Generation Using N-gram Statistics 316
Eder Miranda de Novais, Thiago Dias Tadeu, and Ivandré Paraboni

Text-to-Text Surface Realisation Using Dependency-Tree Replacement 326
Eder Miranda de Novais, Thiago Dias Tadeu, and Ivandré Paraboni

A Comparative Analysis of Centering-Based Algorithms for Pronoun Resolution in Portuguese 336
Fernando José Vieira da Silva, Ariadne Maria Brito Rizzoni Carvalho, and Norton Trevisan Roman

Specification and Evaluation of a Spanish Conversational System Using Dialogue Models 346
Ivan V. Meza, Lisset Salinas, Esther Venegas, Hayde Castellanos, Alejandra Chavarría, and Luis A. Pineda

A Half-Way Semantics toward Collaborative Behavior in Interagent Dialogues 356
M. Julieta Marcos, Marcelo A. Falappa, and Guillermo R. Simari

Using Sentence Semantic Similarity Based on WordNet in Recognizing Textual Entailment 366
Julio J. Castillo and Marina E. Cardenas

Complementing RRL for Dialogue Summarisation 376
Norton Trevisan Roman and Ariadne Maria Brito Rizzoni Carvalho

A Multi-dimensional Annotation Scheme for Behaviour in Dialogues 386
Norton Trevisan Roman and Ariadne Maria Brito Rizzoni Carvalho

Neural Networks

Evolving Artificial Neural Networks Using Adaptive Differential Evolution 396
Adenilton J. da Silva, Nicole L. Mineu, and Teresa B. Ludermir

A Recurrent Neural Network for Channel Assignment Problems in Mobiles 406
Gracián Triviño, José Muñoz, and Enrique Domínguez

SVM Based Feature Selection: Why Are We Using the Dual? 413
Guillermo L. Grinblat, Javier Izetta, and Pablo M. Granitto

Pattern Recognition Using Spiking Neurons and Firing Rates 423
Roberto A. Vázquez

Planning and Scheduling

Interaction Graphs for Multivariate Binary Data 433
Johan Van Horebeek and Jesús Emeterio Navarro-Barrientos

Probabilistic Reasoning

Detecting Drifts in Multi-Issue Negotiations 443
Gislaine M. Nalepa, Bráulio C. Ávila, Fabrício Enembreck, and Edson E. Scalabrin

A Logic Based Algorithm for Solving Probabilistic Satisfiability 453
Marcelo Finger and Glauber De Bona

On-Line Viscosity Virtual Sensor for Optimizing the Combustion in Power Plants 463
Pablo H. Ibargüengoytia and Miguel Angel Delgadillo

Evaluating an Affective Student Model for Intelligent Learning Environments 473
Yasmín Hernández, Luis Enrique Sucar, and Gustavo Arroyo-Figueroa

Dynamic Reward Shaping: Training a Robot by Voice 483
Ana C. Tenorio-Gonzalez, Eduardo F. Morales, and Luis Villaseñor-Pineda

A Visual Grammar for Face Detection 493
Augusto Meléndez, Luis Enrique Sucar, and Eduardo F. Morales

Recognition of Actions That Imply Movement by Means of a Mobile Device with a Single Built-in Accelerometer 503
César Bernardini

Development of a Tour–Guide Robot Using Dialogue Models and a Cognitive Architecture 512
Héctor Avilés, Montserrat Alvarado-González, Esther Venegas, Caleb Rascón, Ivan V. Meza, and Luis Pineda

Detection of Multiple People by a Mobile Robot in Dynamic Indoor Environments 522
José Alberto Méndez-Polanco, Angélica Muñoz-Meléndez, and Eduardo F. Morales-Manzanares

A Motion Planner for Finding an Object in 3D Environments with a Mobile Manipulator Robot Equipped with a Limited Sensor 532
Judith Espinoza and Rafael Murrieta-Cid

A Driving Assistance System for Navigation in Urban Environments 542
Leandro C. Fernandes, Maurício A. Dias, Fernando S. Osório, and Denis F. Wolf

An Efficient Strategy for Fast Object Search Considering the Robot's Perceptual Limitations 552
Javier Cabanillas, Eduardo F. Morales, and Luis Enrique Sucar

Search

Building Feature-Based Maps with B-splines for Integrated Exploration 562
Alfredo Toriz P., Abraham Sánchez L., Rene Zapata, and Maria A. Osorio

A Heuristic Approach for Constructing Ternary Covering Arrays Using Trinomial Coefficients 572
Jorge Martinez-Pena, Jose Torres-Jimenez, Nelson Rangel-Valdez, and Himer Avila-George

Controlling Search in Constrained-Object Models 582
Ricardo Soto

Semantic Web

Adaptivity and Interoperability in e-Learning Using Ontologies 592
José dos Reis Mota and Márcia Aparecida Fernandes

Author Index 603

Building and Assessing Intelligent Tutoring Systems with an e-Learning 2.0 Authoring System

M.L. Barrón-Estrada[1], Ramón Zatarain-Cabada[1], Rosalío Zatarain-Cabada[1], Hector Barbosa-León[2], and Carlos A. Reyes-García[3]

[1] Instituto Tecnológico de Culiacán, Juan de Dios Bátiz s/n, Col. Guadalupe, Culiacán Sinaloa, 80220, México
[2] Instituto Tecnológico de Colima, Calle Av. Tecnológico #1, Villa Alvarez, Colima, 28976, México
[3] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) Luis Enrique Erro No. 1, Sta. Ma. Tonanzintla, Puebla, 72840, México
{rzatarain,lbarron,rcabada}@itculiacan.edu.mx, hbarbosa@itcolima.edu.mx, kargaxxi@inaoep.mx

Abstract. Knowledge Societies also named Social Learning Networks (SLN) allow interaction and collaboration between individuals (instructors and students), who share their connections under a scheme of learning communities around common learning interest. In this paper, we present Zamná, a Knowledge Society implemented as an adaptive learning social network. A community of Instructors and Learners can create, display, share and assess communities, intelligent tutoring systems or adaptive courses in a collaborative environment. The communities and courses are tailored to the student's learning style according to the learning style model of Felder-Silverman. The identification of community's and student's learning style is performed using self-organizing maps. The main contribution of this paper lies at the integration of Artificial Intelligence with SLN.

Keywords: Adaptive mobile learning, Social learning networks, Authoring tools, Learning Styles.

1 Introduction

Social networks sites (SNS) have become a social revolution within web communities, especially young users. They are specifically used by user communities for exchanging and sharing various components such as videos, news, pictures and more. Within these applications, we find that in the area of education SNS have also become a success. Social Learning Networks (SLN) allow interaction and collaboration between individuals (instructors and students), who share their connections under a scheme of communities around common learning interest. The main benefit of using this new approach in education is collecting the information and ideas to create the store of the courses from the whole community of users (including students and instructors). Another benefit is taking advantage of technologies (wikis, blogs and social networking) that young learners are using in their spare time.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 1–9, 2010.

LearnHub, Wiziq, and LectureShare are examples of SLN or e-Learning 2.0 applications (e-Learning software which applies Web 2.0 technologies like social networks, blogging, or wikis) [1]. These SLN provide an online education site for instructors and learners of all kinds. These users can create communities, share courses and lessons, have discussions, make quizzes, etc. However, learning material (courses, lessons, quizzes or tests) authored and used by the users, does not provide direct customized or intelligent instructions to the learners.

In this paper, we propose a new method to create adaptive and intelligent learning material in a context of a SLN. The method implements and uses a Kohonen Neural Network, trained with a set of courses with different learning styles and later used in the learning network or in mobile devices.

The paper's organization is as follows: In Section 2 we give a comparison to related work. In section 3 we present the main structure of the social network *Zamná*. Results and Discussion are shown in Section 4. Conclusions and future work are discussed in Section 5.

2 Related Work

The boom of social networks within the Web has led to the emergence of social networking sites aimed at the education field. Ivanova in [2] presents an analysis of seven systems of e-learning 2.0: EctoLearning, Edu 2.0, 2.0 eLearningCommunity, LearnHub, LectureShare, Nfomedia, and Tutorom(Eduslide). Some of them can be commonly identified as social learning networks according to their characteristics. EctoLearning hosts libraries of knowledge that can be explored, imported, traded, qualified, modified or shared. In Edu 2.0 we can simulate the operation of an educational institution through the functionality provided by the system. The use of the system starts by recording a school through the website, thus generating an online portal of the institution where teachers, students and parents can have access. The site allows teachers to create online classes and students to join them. Within each class the teacher creates lessons that can make use of images, video and audio. It also has the ability to assess and monitor the tasks assigned to students. WiZiQ is completely free online software, which allows users to interact through text, audio and video with other participants in a shared virtual space. The space contains a whiteboard with powerful drawing tools, where users move between different boards and download PowerPoint presentations to show one or more users.

All these systems have multiple functionalities to enable an author to create, deploy, share and evaluate different types of learning objects. However, learning objects deployed to teach and evaluate all students are the same. This means that there is no recognition of individuality in the way students learn and are evaluated.

Related work for adapting learning styles have been implemented using different approaches like Bayesian networks, decision trees, or Hidden Markov Models [3, 4, 5]. In those works, learning styles are calculated based only on the ILSQ questionnaire and none of them are authoring tools.

3 The SLN Structure

Figure 1 illustrates the structural design of Zamná. A user enters the site (Zamná) using any browser. Then, the user can create courses, communities and lessons, upload documents, make contacts (friends, etc.). The user can build adaptive courses (intelligent tutoring systems) for students who learn under different learning styles according to the Felder-Silverman Learning Styles Model [6]. We can also observe in figure 1 that a course created with Zamná can be exported to a mobile device or browsed on the site. A community or knowledge society contains, among other intelligent tutors, lessons and educational materials used by this community. For example, a knowledge society of Compilers contains all the learning materials that users of this society upload, create, share and study. The courses and communities contain sections of assessments, which are also carried out according to the best student's learning style.

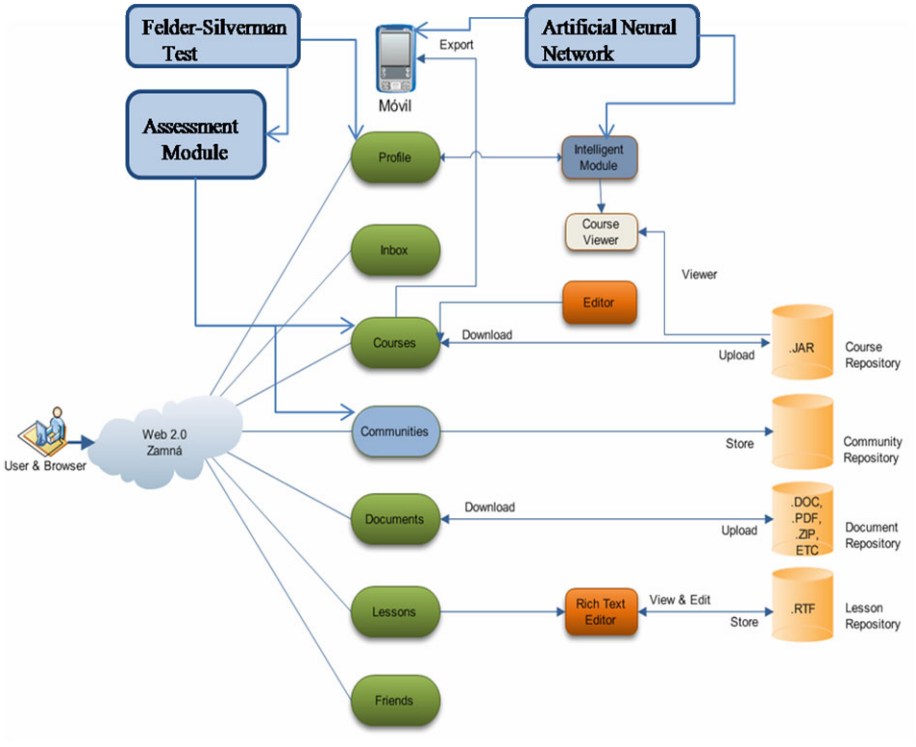


Fig. 1. Zamná Structural Design

With respect of the society evaluation, the goal is to increase their knowledge in a particular area of study by positioning them in a new and higher level. The current method to gather information from a community is through the publication of individual tests with adaptive characteristics for each student. This instrument is an objective and formative test to give feedback to the individual and evaluate the

group´s outcome. We use the user´s learning style identified in Zamná to define the first adaptation process. We use also adaptation techniques and methods to construct adaptive tests by grouping items according the associated learning material. To define the adaptive test we developed multiple versions of a single question which present alternative multimedia material (i.e. audio, video or text) according to the users' learning style identified previously in Zamná. Also we categorize those items according a complexity level to perform a second adaptation process. Once the student accomplished a learning activity in Zamná he/she performs a formative test to measure the acquired knowledge and contribute to the group´s outcomes.

Figure 2 shows different sections of the Zamná Environment. In clockwise order we observe 4 different interfaces for creating a community, uploading a course, showing a quiz, and browsing a course.

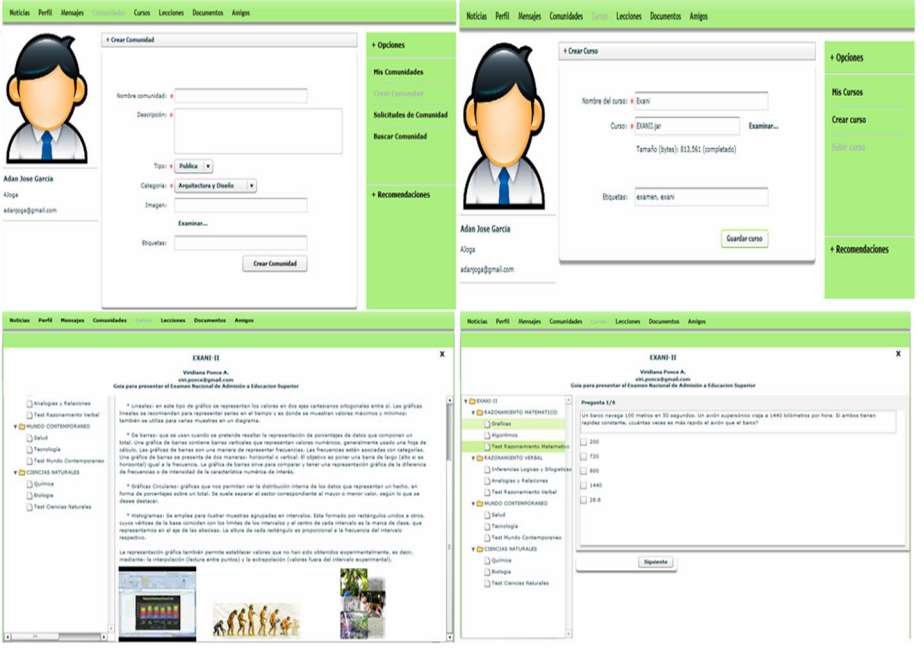


Fig. 2. Zamná Environment

3.1 The Intelligent Module

The Intelligent Module consists of a predictive engine used to identify dynamically the student´s learning style whenever a tutoring system is running. At the beginning an interpreter selects content (learning objects) based upon the student´s learning style obtained from the student profile. The learning style of a student can be modified according to evaluations applied to the student. The process of constructing a new tutoring system (an adaptive or intelligent system) consists of three main steps.

During Step 1 a tree structure of the adaptive or intelligent tutoring system is designed by the authors of the learning material. On the tree structure, the designer

also inserts quizzes. Quizzes are an important element to provide adaptation capabilities to the produced tutors. In step 2 the tree structure is filled with domain contents (a knowledge base), and some other learning resources. At the beginning of the creation the author creates the tutoring system by inserting different learning objects like text fragments, images, audio/video clips, and by defining learning styles, prerequisites, tags and quizzes.

At a later time, learning resources are added to the tutoring system by learners, who recommend resources they find commonly on the web. Those resources can be found also in a special resource repository. After the author creates the tutoring system, she/he can save it and export it to be browsed or displayed in the Learning site or in mobile devices (step 3). The saved/exported file will enclose three elements: an XML file corresponding to the learning resources or contents, a predictive engine for navigation purposes**,** and a Kohonen Neural Network [7] for learning style classification.

The predictive engine dynamically identifies the student´s learning style whenever he/she is running the tutoring system. At the beginning, an interpreter selects content (learning objects) based upon the student´s learning style, obtained from the student profile. The learning style can be modified according to evaluations applied to the student. The engine makes use of an Artificial Neural Network, a Self-Organizing Map or SOM, for selecting student´s learning styles. We decided to use a SOM because it is an unsupervised network, and this allows us to avoid the support of experts in education to build a desired output for each combination of learning styles. The input layer of the neural network has 7 neurons. The Kohonen layer in the SOM has 1600 neurons, organized in a lattice with hexagonal cells with dimensions of 40x40 neurons. The signals are part of the training data space, and they are vectors composed of three elements: two vectors and a scalar. The first vector is the student's learning style obtained from a test: The Questionnaire Learning Styles Inventory of Felder-Solomon [8] applied to a group of high school students. The second vector is the learning style with which we designed the learning material studied by the group of students. The last element (a scalar) is the performance of the student who has a learning style obtained from the Felder-Solomon test and is studying a course designed with one of the possible learning styles. We created learning material for three different subjects or courses: photography, eolic energy and introduction to computers. For each of these courses we created 8 different versions. Each version was modeled using a different learning style. We implemented 3 dimensions of the Felder-Silverman model, with a result of 8 different learning styles (visual-verbal, sensitive-intuitive and sequential-global dimensions). Once the neural network is successfully trained, it may be used to identify learning styles of students.

3.2 Editing the Knowledge Base for a Tutoring System

We apply the Knowledge Space Theory [9]. This theory gives a sound foundation for representing the knowledge domain for adaptive (intelligent) learning. An adaptive assessment based upon knowledge spaces and student prerequisites (and employing a Kohonen neural network for identifying learning styles), will derive a particular or personalized path of learning objects. For each topic a set of prerequisites and quizzes are set. Figure 3 shows the knowledge domain of the topic *parsing* or syntactic analysis for a compiler construction course. A Dashed line represents a personalized path.

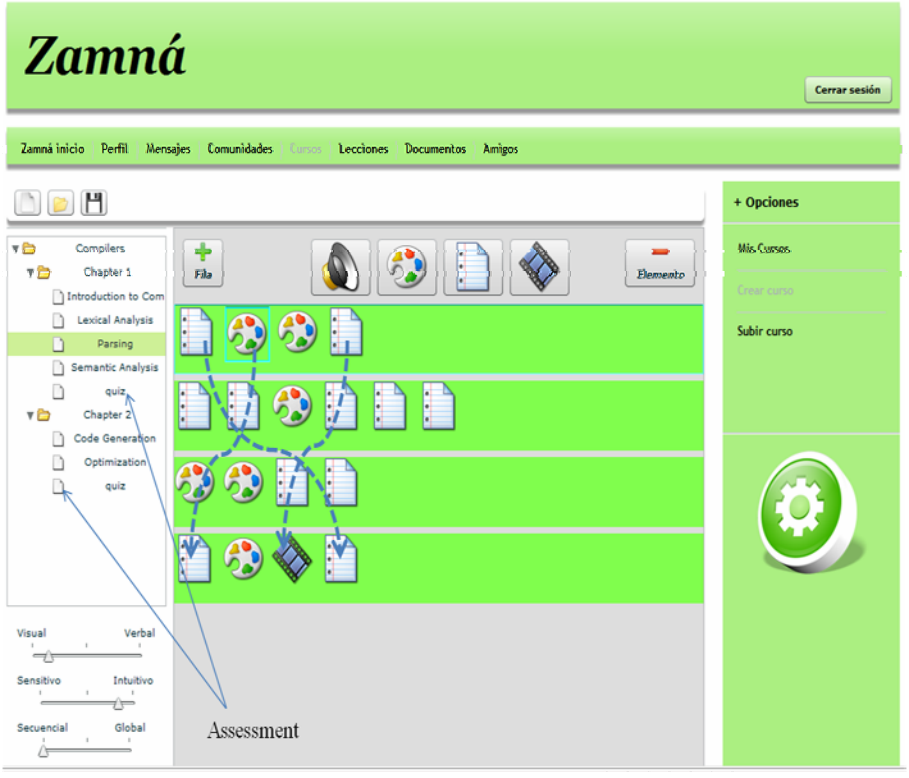


Fig. 3. Knowledge Domain for the parsing topic

3.3 Assessment Module for Students and Knowledge Societies (Communities)

In figure 3, we can observe two evaluation elements included in the course of compilers. The results of these assessments applied to each student will help the neural network to dynamically assign the best student's learning style. Assessments are also made according to learning style. So that the student can study, learn and be evaluated according to a learning style.

Moreover in addition to the individual questionnaires in Zamná, the tool also allows you to design and develop community-adaptive tests or assessments [10] to enable members of a society or community to measure their knowledge in a particular topic, encouraging self-learning and growth throughout a community identified as a "knowledge society." The community assessment is currently implemented with only the scale of input (visual/verbal). Each test to the community is based on only two different learning styles. The test result allows us to decide whether the community learns more by studying verbal or visual learning material. We use also the SOM neural network to change the learning style of the whole community.

Next, we present the algorithm to identify the student´s learning style.

I. The SOM receive D = [e r] as data input, where *e* is the learning style of the learning material presented by the student in Zamná, and *r* is the current score of the student which has a predefined learning style with the Felder-Solomon test. In the case of a community assessment *e* only takes the scale of input (visual/verbal).
II. A search for the BMU (best matching unit) that contains a weight vector W = [E r e] where *E* is the Felder-Solomon student´s learning style in the space of the Kohonen network is made, to find the neuron that best matches the input vector.
III. The network returns the new student´s or community´s learning style *E*.
IV. The algorithm repeats steps I, II, and III whenever a test to a student or to the community is performed, and it is done when evaluation has ended.

Fig. 4. Example of a Community (System Programming)

4 Results and Discussions

The tool has been tested developing communities, courses, etc. Figure 4 shows the community "systems programming" (programación de sistemas) which includes members (students and teachers), intelligent tutors, resources used (software, documents, data files, etc.). Zamná has been used to build intelligent tutors on issues such as basic mathematics, the Maya language, compiler construction, etc. One of the features of the website of the tool is that when viewing an intelligent tutor, you can also display a log of the learning styles since the student has taken the

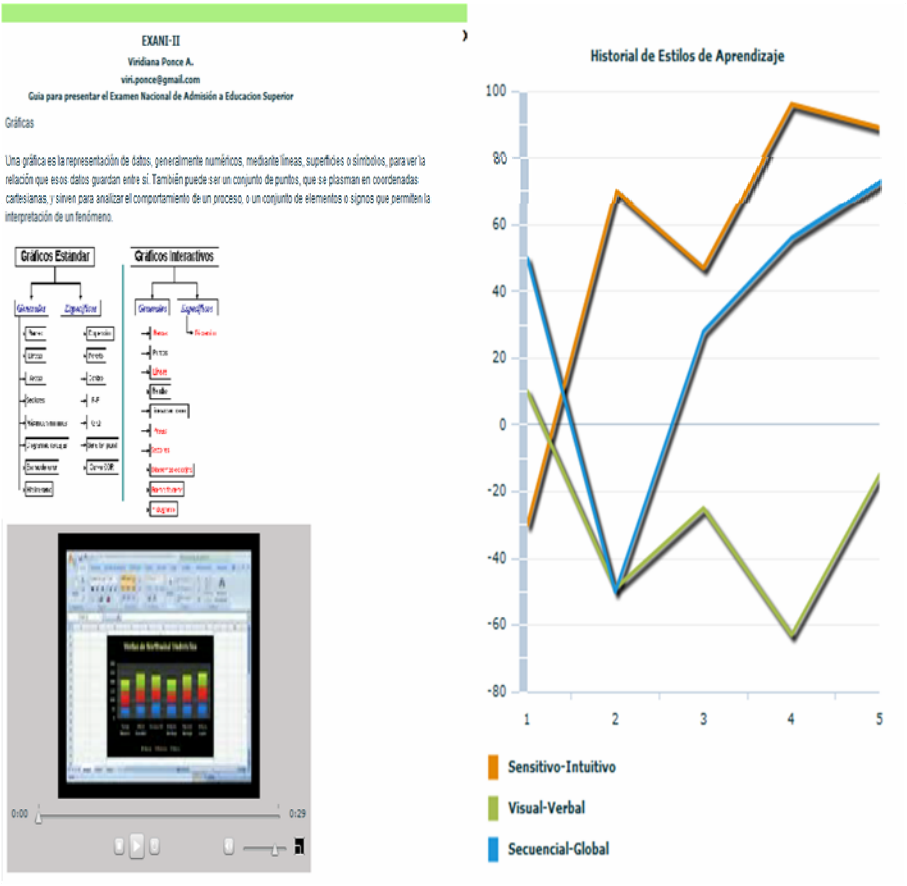


Fig. 5. An Intelligent System (left) and the Learning Style Log (right) of a Student taking the Course

Felder-Solomon test, passing through the different styles the neural network calculates. Figure 5 illustrate part of a Tutoring System (left) and the graphic for the learning styles the system calculates (right).

5 Conclusions and Future Work

The design of the graphical user interface (presentation layer) was developed using Adobe Flex technology. The implementation of the classes (business layer) used in the interface was developed with the PHP programming language. The data transmission between the two layers is through standard XML markup language. The web server is Apache and the Database Management System is MySQL. We can reach the SLN Zamná in *http://www.posgradoitc.edu.mx/zamná/*.

We have plans to add new components to the editor, for example we want to add the dimension of processing (active / reflective), which requires components with

focus on individual work (active) or group (reflective). For example we want to add forums, blogs, chats, etc. With them we will strengthen the reflective learning.

Acknowledgments. The work described in this paper is fully supported by a grant from the DGEST (Dirección General de Educación Superior Tecnológica) in México.

References

1. Boyd, D.M., Ellison, N.B.: Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication 13(1), article 11 (2007)
2. Ivanova, M.: Knowledge Building and Competence Development in eLearning 2.0 Systems. In: I-KNOW 2008, Graz, Austria, September 3-5, pp. 84–91 (2008)
3. Carmona, C., Castillo, G., Millán, E.: Designing a Bayesian Network for Modeling Student's Learning Styles. In: Díaz, P., Kinshuk, A.I., Mora, E. (eds.) ICALT 2008, pp. 346–350. IEEE Computer Society, Los Alamitos (2008)
4. Graf, S., Kinshuk, A.I., Liu, T.: Identifying Learning Styles in Learning Management Systems by Using Indications from Students' behavior. In: Díaz, P., Kinshuk, A.I., Mora, E. (eds.) ICALT 2008, pp. 482–486. IEEE Computer Society, Los Alamitos (2008)
5. Limongelli, C., Sciarrone, F., Vaste, J.: LS-PLAN: An Effective Combination of Dynamic Courseware Generation and Learning Styles in Web-based Education. In: Nejdl, W., Kay, J., Pu, P., Herder, E. (eds.) AH 2008. LNCS, vol. 5149, pp. 133–142. Springer, Heidelberg (2008)
6. Felder, R.M., Silverman, L.K.: Learning and Teaching Styles in Engineering Education. Engineering Education 78, 674–681 (1988)
7. Kohonen, T.: Self-Organization and Associative memory, 3rd edn. Springer, Heidelberg (1989)
8. Felder, R.M., Solomon, B.A.: Index of Learning Styles Questionnaire, http://www.engr.ncsu.edu/learningstyles/ilsweb.html
9. Doignon, J.-P., Falmagne, J.C.: Knowledge Spaces. Springer, Heidelberg (1999)
10. Wolf, P., Hill, A., Evers, F.: Handbook for Curriculum Assessment. University of Guelp, Ontario (2006)

Robotic Orientation towards Speaker for Human-Robot Interaction

Caleb Rascón, Héctor Avilés, and Luis A. Pineda*

Instituto de Investigaciones en Matemáticas Aplicacas y en Sistemas (IIMAS)
Universidad Nacional Autónoma de México (UNAM)
{caleb,haviles}@turing.iimas.unam.mx, luis@leibniz.iimas.unam.mx

Abstract. The orientation of conversational robots to face their interlocutors is essential for natural and efficient Human-Robot Interaction (HRI). In this paper, progress towards this objective is presented: a service robot able to detect the direction of a user, and orient itself towards him/her, in a complex auditive environment, using only voice and a 3-microphone system. This functionality is integrated within Spoken HRI using dialogue models and a cognitive architecture. The paper further discusses applications where robotic orientation benefits HRI, such as a tour-guide robot capable to guide a poster session and a "Marco Polo" game where a robot aims to follow a user purely by sound.

Keywords: Cognitive architecture, Human-robot interaction, Direction-of-arrival estimation, Robotic orientation.

1 Introduction

The orientation of a service robot is essential for a natural Human-Robot Interaction (HRI). From the user experience point of view, the human will feel the interaction as more 'natural' if the robot is facing him/her during the conversation. However, as such ability requires the direction of the user to be known, it can benefit other parts of HRI.

For instance, once the direction of the user is known, voice recognition can be improved using directional noise cancellation [7]. In addition, it is well known that face detection and recognition provide rich information relevant to HRI: the identity of the user, the direction the user is looking at, his mood, etc. [4,18] However, such analysis is carried out by visual means, and the user cannot always be expected to be in the line of sight. Knowing the direction of the user by sound alone, and facing the user accordingly, tackles this issue straight on. Moreover, it is imperative to know the direction of the user in a situation such as the one proposed in the test *Follow me*, taking place in the popular *Robocup@Home* competition[1], where the robot is required to track and follow the user.

* The authors would like to acknowledge the support from the grants CONACYT 81965 and PAPIIT-UNAM IN-104408 and IN-115710.

[1] www.robocup2010.org

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 10–19, 2010.

Knowing the direction of the user in regard to the robot is also a good first step towards localizing him/her in the environment. In a 3-dimensional polar coordinate system, the horizontal angle (i.e. the direction of the user) is one of three values that define a location (the other two being: vertical angle and distance from origin). And the location of the user is, in turn, another important variable in HRI. During a human-robot conversation, the phrase "robot, come here" may be emitted by the human. In this situation, even with the phrase recognized correctly, the robot may know that it needs to move, but, because the term 'here' lacks context, it will not know **where** to move.

Unfortunately, locating the user in the environment is a difficult task to carry out by sound analysis alone. It is of interest having several types of modalities (vision, sound, laser, etc.) working in conjunction for that objective [14,19,13], where the direction of the user plays an important role [12].

In addition, many types of noise may hinder the estimation of the sound source Direction-of-Arrival (DOA), such as reverberation [2]. A sophisticated recording system may be able to overcome them, such as the one proposed in [17] that used a 24-microphone 1-D array for precision. However, a high amount of microphones may be impractical to carry by many of the currently-in-use service robots [20,11], such as our in-house robot, Golem, herein described.

Golem is a service robot built with a primary focus on HRI. It is integrated by a cognitive architecture focused on HRI, explained in detail in [14], which can take advantage of different types of information interpreted from the world, including the direction of the user. Because Golem is a conversational robot, it is of interest to implement a robotic-orientation module that is triggered by sound. This implies that the module needs to be sufficiently light on the hardware side for the robot to carry, but robust enough in the software side to handle different types of noise and disturbances. Moreover, for tracking purposes, the module should be able to estimate the direction of the user in a -179°– 180° range, and fast enough to do so with small utterances from the user.

The paper is organized as follows: Section 2 further discusses how a direction-aware service robot may benefit HRI, as well as give a brief review of current algorithms that aim to estimate the direction of a speaker; Section 3 describes the hardware setup and the proposed algorithm; the different trials on an actual service robot and their results are given in Section 4; and conclusions are discussed in Section 5.

2 Background

2.1 On Benefits to Human-Robot Interaction

A cognitive architecture for a service robot, designed to focus primarily on HRI, is presented in [14], and is implemented in our service robot Golem. This cognitive architecture has a top level, called the 'Representation & Inference' phase, where the set of expected multimodal situations are defined and ordered in what is called a Dialogue Model (DM), which guides the HRI process and assumes an abstract-but-meaningful interpretation of the world. This interpretation is

achieved by first obtaining an uninterpreted image [14] (i.e. the Recognition phase), and then providing a meaning to that representation (i.e. the Interpretation phase), guided by the contents of the episodic memory of the architecture and the current expectations of the DM. With this architecture, complex objectives can be fulfilled, such as a tour-guide robot capable to guide a poster session [1] and a spanish-spoken "guess the card" game [8].

An example of a DM is shown in Figure 1, which describes the popular Marco Polo game[2]. The HRI process flows through the DM, with each node describing a situation. Each node is joined to other nodes by arrows that are tagged with an "Expectation:Action" label. If the Expectation with which an arrow is tagged is met, the corresponding Action is carried out and the system moves to the node the arrow is pointing at.

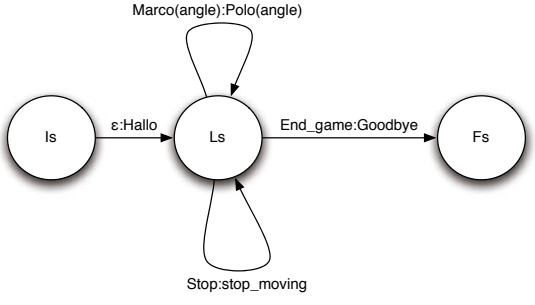


Fig. 1. Dialogue Model for the Marco Polo game

An Expectation is met when the abstract interpretation of the world matches the expected occurrence in the world, such as the user greeting the robot at the start of a tour or the robot bumping into something when moving. If an expectation is met accordingly, the DM furthers along the HRI process; if something happens in the world, but no expectations were in place for it in the current situation in the DM, the system executes a recovery dialogue model to continue the HRI.

In the example of Figure 1, the expectation *Marco(angle)* is met when the user specifically says the word "Marco", detected to be coming from the direction *angle*, and it triggers the action *Polo(angle)*, indicating to the robot to stop, say the word "Polo", turn to the direction *angle*, and continue moving. The expectation *Stop* is met when the user expresses the wish for the robot to stop, and it triggers the action *stop_moving*, indicating to the robot to stop. Finally, the expectation *End_game* is met when the user expresses the wish for the game to end, and it triggers the action *Goodbye*, indicating to the robot to stop and express a farewell.

[2] A game between two parties, where one is blind and tries to follow the other party by sound alone. The followed party must yell "Marco" to signify where he/she is, and the following blind party must respond "Polo" and move to that location.

The module proposed in this paper will reside inside the Recognition phase, where the sound of the outside world will be tagged with the characteristic of "direction". This characteristic can be used by the Interpretation phase to enrich the meaning of the information received by other modalities, such as in the example presented in [1], where the tour-guide robot could approach and greet the user at the start of the tour, triggered by sound alone (instead of manual means); the robot could also face the user when being talked to during the tour, enhancing the 'naturality' of the conversation. The "direction" characteristic may also be used directly by the DM, as in the case of the Marco Polo example, where such information is essential for carrying out HRI. To do any of these, however, a sound source direction estimator needs to be implemented first.

2.2 On Source Direction Estimation

Estimating a Sound Source Direction of Arrival (DOA) is a well written-about topic in Signal Processing. It has been proven useful in applications ranging from fault monitoring in aircrafts [17], to intricate robotic pets [6], to close-to-life insect emulation [5]. In addition, the principles employed in DOA estimation have been applied in the design of hearing aids [7].

Having two audio sensors (i.e. microphones), the Inter-aural Time Difference (ITD) is the delay of a sound from one microphone to the other. It is one of the main sources of information for DOA estimation, as it provides a clear relation to the direction of the source (it was applied in [10] with limited results). The Inter-aural Intensity Difference (IID) is the difference in magnitude between both microphones and can also be used for DOA estimation, although a training stage is usually necessary for it to be useful, as it was observed in [11].

In [2], the concept of Inter-aural Coherence (IC) is introduced, which is the highest correlation measure of the cross-correlation vector between the two signals. If a high IC is present, the signals are deemed coherent and, thus, an analysis using ITD and/or IID can proceed. This methodology was implemented in [6], and it was observed that it didn't improve DOA estimation when dealing with complex signals (e.g. more than one source, reverberation present, etc.).

The Multiple Signal Classification algorithm (MUSIC) [16] is able to detect the Direction of Arrival (DOA) of as many sources as one less the available microphones (e.g. 1 source for 2 microphones). It does this by projecting the received signals in a DOA subspace, based on their eigenvectors, similar to Principal Component Analysis. It was applied in [9] with good results, although it has been observed that its performance decreases considerably in the presence of reverberation [21] (pp. 169).

Classic beamforming can be applied for precise DOA estimation [17], but requires a large quantity of microphones which is impractical for smaller robots.

The DOA of a source has been able to be estimated using one microphone by implementing an 'artificial ear' [15], but the sound was required to be known *a priori* and any modification to the ear (even its location in relation to the microphone) required re-training.

More specifically in the area for service robots and HRI, DOA estimation can be considered to be in its initial stages. In [20], the sources are separated from each other, in order to enhance speech recognition, and as a preamble for DOA estimation. However, it required an array of 8 microphones positioned in a cube-like manner to work, which may be impractical for service robots.

Other attempts, such as [12], have approached the DOA estimation problem using a two-microphone array. The reasoning behind using only two microphones ranges from that of practicality (it is lightweight) to that of biological similarity [21], where the robot is meant to be the most human-like possible [3]. However, doing so comes with three main problems.

The feature most used for DOA estimation when using a 2-mic array is the ITD, as there is a direct relation between the two, described in equation (1).

$$\theta = arcsin\left(\frac{ITD \cdot V_{sound}}{F_{sample} \cdot d}\right) \tag{1}$$

where θ is the DOA angle; ITD is the Inter-aural Time Difference in number of samples; V_{sound} is the speed of sound ($\sim$ 343 m/s); F_{sample} is the sampling frequency (usually 44.1 kHz); and d is the distance between microphones.

In Figure 2, the DOA is plotted against the ITD, and it can be seen that in the -50°–50° range, the relation between both seem almost linear. However, in the outer ranges, the relation becomes exponential. This causes major errors when estimating angles that are located in the sides of the robot [12].

As it can also be seen in Figure 2, a 2-mic array only estimates DOAs in the -90°– 90° range. This can be surmounted by implementing 'artificial ears' that can detect if the sound source is coming from the front or back of the robot, but it has been proven impractical [15]. This can also be tackled by a two-phase strategy: a first pair of signals are used to estimate an initial DOA, the robot then rotates briefly, and then another pair of signals are acquired to estimate a second DOA. Comparing both DOAs provides an angle in the -179°– 180° range,

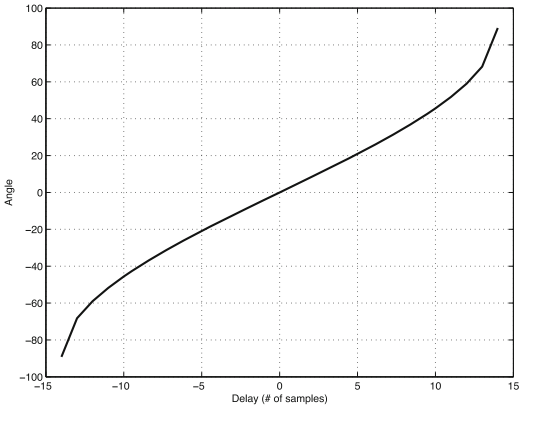


Fig. 2. DOA (or *Angle*) versus ITD (or *Delay*) in number of samples

but has its own set of issues: it requires considerably more time than when using one DOA estimate, the required rotation may hinder navigational requirements, and the user may be moving as well, rendering the DOA comparison mute.

Finally, the estimation of the ITD is usually based on the calculation of the cross-correlation vector between the two signals. This process can be very sensitive to reverberations and other noise sources when using only 2 microphones [21] (pp. 213-215), which may result in significant errors in the DOA estimation.

3 A 3-Microphone System

To avoid the problems described in the last section that arise when estimating a DOA using only two microphones, and to maintain a relatively light hardware setup, a three-microphone system, as it is shown in Figure 3, is proposed.

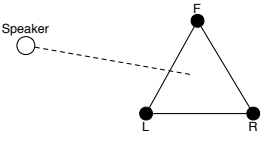


Fig. 3. The proposed 3-microphone system

The distances RL, LF, and FR are equal, thus a preliminary angle can be calculated using equation (2).

$$\theta_m = arcsin\left(\frac{ITD_{min} \cdot V_{sound}}{F_{sample} \cdot d}\right) \tag{2}$$

where ITD_{min} is the ITD with the lowest absolute value of the three ITDs. Equation (3) is then used to calculate the final DOA value (θ).

$$\theta = f(\theta_m, ITD_{RL}, ITD_{LF}, ITD_{FR}) \tag{3}$$

where f is a function that first checks if the three ITDs (ITD_{RL}, ITD_{LF}, and ITD_{FR}) point to the same angle sector. It is a type of redundancy check against situations with reverberation and/or with more than one sound source. If the ITDS are redundant, f shifts θ_m to the pointed angle sector to obtain θ.

With this setup, θ is always estimated using a θ_m inside the -30°– 30° range (well within the close-to-linear -50°– 50° range), resulting in a close-to-linear ITD-DOA relation all through the -179°– 180° range.

In addition, because of the redundancy check and the close-to-linear relation, the maximum error of this system can be known beforehand using equation (4).

$$|error^{\circ}_{max}| = arcsin\left(\frac{ITD_{>30^\circ} \cdot V_{sound}}{F_{sample} \cdot d}\right) - arcsin\left(\frac{ITD_{<30^\circ} \cdot V_{sound}}{F_{sample} \cdot d}\right) \tag{4}$$

where $ITD_{>30^\circ}$ and $ITD_{<30^\circ}$ are the ITDs (measured in number of samples) that provide the closest ceil and floor measurements, respectively, to 30°. In the case of Golem, sampling at 44.1 kHz and with the microphones spaced at 18 cm, a maximum error of ±2.8747° can be expected. In those same circumstances, when using a 2-Mic array, the maximum expected error is ±15.0548°, which occurs when the sound source is close to either side of the robot.

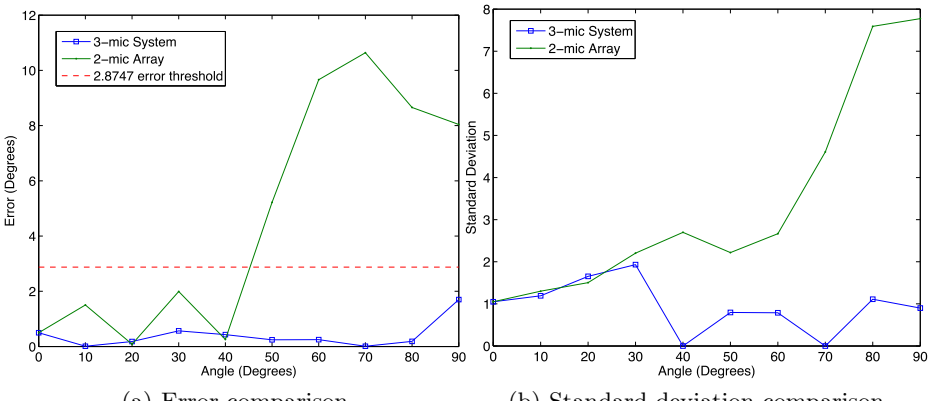
4 Trials and Results

Two trials were implemented: one offline, where the angle is estimated in a relatively quiet environment, and another online, where the angle was used as part of a simple navigation system implemented in Golem.

The offline trial was carried out to compare the proposed 3-mic system and a 2-mic array in a situation where both systems can potentially provide good results. Because a 2-mic array is not able to detect angles outside the -90°– 90° range, 10 equally spaced angle values within the 0°– 90° range were used as a testbed for both systems (the -90°– 90° range is symmetrical).

At each angle value, a sound source was located at 20 cm from the system, and 10 sets of 4410 samples were taken. With each sample set, an angle was estimated; the average of the 10 estimated angles was considered as the final estimated angle. The final estimation was then subtracted from the known angle to calculate the error. The standard deviation of the 10 angles was also calculated. The results of the trial are shown in Figure 4.

As it can be seen in Figure 4a, the 3-mic system is able to provide good angle estimates throughout the angle range, while the 2-mic array error increases substantially with angles greater than 40°. In addition, it can be seen in Figure 4b that the 2-mic array, when estimating large angles, estimations deviate considerably more from one moment to another than the 3-mic system. This results show that the 3-mic system not only provides better angle estimations than the



(a) Error comparison (b) Standard deviation comparison

Fig. 4. Results of offline trial

two 2-mic array, but also that the estimations are more consistent throughout the angle range. Adding on the fact that this consistency is provided in all of the -179°– 180° range, this results show that 3-mic system is superior in all respects to the 2-mic array, while being comparably lightweight.

Because the 3-mic system overly outperformed the 2-mic array, it was meant appropriate to install over Golem, which triggered the online trial. The testing area was a large room, with higher reverberation than that of the offline trial, and in presence of higher ambient noise (fans, keyboard typing, low-to-medium chatter between desks). A simple navigation system was implemented, such that when an angle is provided, the system would turn towards that angle. The sound source was a user who was instructed to utter the phrase "Golem" from 9 different angular positions 2 meters away from the robot. All estimations were carried out starting from the 0° position and only the 3-mic system was tested[3]. The results are shown in Figure 5.

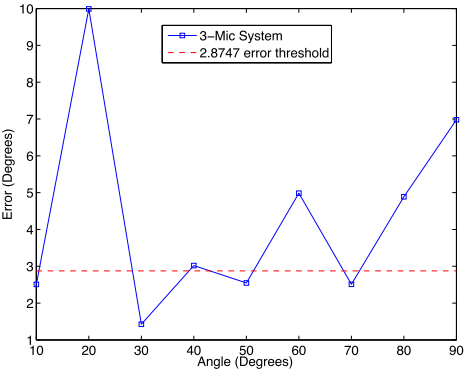


Fig. 5. Results of online trial

As it can be seen, the estimations are considerably worsened in the online trial. However, in all cases, except for the 20° and 90° position, the estimation is no more than 3° above the offline error threshold, which for real-life use is more than adequate. In fact, the user that was testing the robot regarded its orientation as adequate in all the angular positions, except at 20°. The error spikes in the 20° and 90° positions are currently being investigated.

It is important to note that this was a real-life test. The sonic complexity of the room was high, the user placement can be expected to be inconsistent, the navigation system could have produced some positional errors, and no reverb filtering was carried out. When considering all of this, the 3-mic system has shown it is an adequate solution to the user DOA estimation problem.

[3] The 2-mic array was deemed inappropriate for this scenario. It consistently misidentified the sound source, making the robot behave erratically.

5 Conclusion and Future Work

Human-Robot Interaction benefits from a rich perception of the world. Having the robot orient itself towards the user during a conversation enhances HRI from the point of view of both the user and the robot: the 'naturality' of the conversation is improved, and the acquisition of more information from the user (face recognition, voice context, etc.) is simplified. To do this, however, the direction of the user is required. Because a conversation is carried out via voice, it is appropriate that the direction of the user be estimated by sound analysis.

It was shown that a 3-microphone system provides a more reliable Direction-of-Arrival estimation than the more-widely-used 2-mic array, while being light enough to be carried by a service robot. It also provided a robust estimation in the presence of reverberation and other types of noise. However, the 3-mic system could only go so far, as the implemented navigation system in Golem (our service robot) contributed some orientation errors that need to be amended.

A good first step was presented towards the integration of a robotic-orientation module in various HRI applications, such as a tour-guide robot and a "Marco Polo" game. This integration is considered as future work in this project.

It is important to note that it is intended for the 3-mic system to be the medium with which the Automatic Speech Recognizer (ASR) gets audio data. Currently, when using the onboard 3-mic system, the performance of the ASR decreases significantly, compared to when using a headset. The reason for this is that, even though the direction estimator is now more robust against reverberation, the ASR is not. To this effect, the next phase of this work is to tackle the reverb problem, aided by the 3-mic system, to enhance voice recognition.

References

1. Avilés, H., Alvarado-González, A.M., Venegas, E., Rascón, C., Meza, I., Pineda, L.A.: Development of a tour–guide robot using dialog models and a cognitive architecture. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 512–521. Springer, Heidelberg (2010)
2. Faller, C., Merimaa, J.: Source localization in complex listening situations: Selection of binaural cues based on interaural coherence. The Journal of the Acoustical Society of America 116(5), 3075–3089 (2004)
3. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots. Robotics and Autonomous Systems 42(3-4), 143–166 (2003)
4. Hjelmås, E., Low, B.K.: Face detection: A survey. Computer Vision and Image Understanding 83(3), 236–274 (2001)
5. Horchler, A.D., Reeve, R.E., Webb, B., Quinn, R.D.: Robot phonotaxis in the wild: a biologically inspired approach to outdoor sound localization. In: Sound Localization, 11th International Conference on Advanced Robotics (ICAR 2003), pp. 1749–1756 (2003)
6. Liu, R., Wang, Y.: Azimuthal source localization using interaural coherence in a robotic dog: modeling and application. Robotica First View, 1–8 (2010)

7. Lockwood, M.E., Jones, D.L., Bilger, R.C., Lansing, C.R., O'Brien Jr., W.D., Wheeler, B.C., Feng, A.S.: Performance of time- and frequency-domain binaural beamformers based on recorded signals from real rooms. Journal of the Acoustical Society of America 115(1), 379–391 (2004)
8. Meza, I., Salinas, L., Venegas, E., Castellanos, H., Chavarria, A., Pineda, L.A.: Specification and evaluation of a spanish conversational system using dialogue models. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 346–355. Springer, Heidelberg (2010)
9. Mohan, S., Lockwood, M.E., Kramer, M.L., Jones, D.L.: Localization of multiple acoustic sources with small arrays using a coherence test. Journal of the Acoustical Society of America 123(4), 2136–2147 (2008)
10. Murray, J.C., Erwin, H., Wermter, S.: Robotics sound-source localization and tracking using interaural time difference and cross-correlation. In: AI Workshop on NeuroBotics (2004)
11. Murray, J.C., Erwin, H.R., Wermter, S.: Robotic sound-source localisation architecture using cross-correlation and recurrent neural networks. Neural Networks 22(2), 173–189 (2009)
12. Nakadai, K., Okuno, H.G., Kitano, H.: Real-time sound source localization and separation for robot audition. In: Proceedings IEEE International Conference on Spoken Language Processing, pp. 193–196 (2002)
13. Pineau, J., Montemerlo, M., Pollack, M., Roy, N., Thrun, S.: Towards robotic assistants in nursing homes: Challenges and results. Robotics and Autonomous Systems 42(3-4), 271–281 (2003)
14. Pineda, L.A., Meza, I., Salinas, L.: Dialogue model specification and interpretation for intelligent multimodal HCI. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 20–29. Springer, Heidelberg (2010)
15. Saxena, A., Ng, A.Y.: Learning sound location from a single microphone. In: ICRA 2009: Proceedings of the 2009 IEEE International Conference on Robotics and Automation, pp. 4310–4315. IEEE Press, Piscataway (2009)
16. Schmidt, R.: Multiple emitter location and signal parameter estimation. IEEE Transactions on Antennas and Propagation 34(3), 276–280 (1986)
17. Smith, M., Kim, K., Thompson, D.: Noise source identification using microphone arrays. Proceedings of the Institute of Acoustics 29(5) (January 2007)
18. Stiefelhagen, R., Ekenel, H.K., Fugen, C., Gieselmann, P., Holzapfel, H., Kraft, F., Nickel, K., Voit, M., Waibel, A.: Enabling multimodal human-robot interaction for the karlsruhe humanoid robot. IEEE Transactions on Robotics 23(5), 840–851 (2007)
19. Tanawongsuwan, R., Stoytchev, A., Stoytchev, E., Essa, I.: Robust tracking of people by a mobile robotic agent. Tech. rep. (1999)
20. Valin, J., Rouat, J., Michaud, F.: Enhanced robot audition based on microphone array source separation with post-filter. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 2123–2128 (2004)
21. Wang, D., Brown, G.J.(eds.): Computational auditory scene analysis: Principles, Algorithms, and Applications. IEEE Press/Wiley-Interscience (2006), http://www.casabook.org/

Dialogue Model Specification and Interpretation for Intelligent Multimodal HCI*

Luis A. Pineda, Ivan V. Meza, and Lisset Salinas

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS)
Universidad Nacional Autónoma de México (UNAM)
luis@leibniz.iimas.unam.mx,
{ivanvladimir,liz}@turing.iimas.unam.mx

Abstract. A methodology and programming environment for the specification and interpretation of dialogue models for grounded multimodal interaction is presented. This conceptual framework permits the declarative specification of complex interactive systems with multimodal input and output, including speech, computer vision and motor behavior. We first introduce the present notion of dialogue model with its motivation on the structure of conversation. Then, the specification and interpretation of dialogue models is presented and discussed. We also present a cognitive architecture for the construction of intelligent Human-Computer Interaction (HCI) applications within this conceptual framework. The paper concludes with references to working systems, demos and work in progress built within the present framework.

Keywords: Dialogue Models, Task Structure, Cognitive Architecture, HCI.

1 Intentional Specification of HCI Protocols

In this paper we present current developments and applications of a methodology and a programming environment for the specification and implementation of intelligent multimodal systems [13]. This framework permits the development of in situ and mobile applications, with spoken language and vision, and other interactive input and output modalities. The approach is motivated by our previous work on the structure of task-oriented conversations and the DIME-DAMSL tagging scheme [12], which was a development on the DAMSL scheme [2] but taking into consideration insights from Clark and Shaefer's theory of conversational contributions and grounding [5], in addition to our own empirical investigation. Following DAMLS, DIME-DAMSL analyzes speech acts at the information, agreement and communication levels; however, following Clark and Shaefer's, speech acts make contributions that need to be "balanced" by other conversational contributions, until the conversation is "grounded", and the conversation as a whole has a stack like structure. The relations between speech acts leads to the definition of conversational protocols that people

* We acknowledge the support of the members of the DIME and Golem group at IIMAS, UNAM. We also gratefully thank the support of grants CONACyT 81965 and PAPPIT-UNAM IN-121206, IN-104408 and IN-115710.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 20–29, 2010.

naturally follow in conversation. The present framework formalizes this notion, and a conversational domain is characterized in terms of a set of generic conversational protocols and set of speech act types that people use to communicate in such a domain. Related work on the specification of multimodal HCI and Human-Robot interaction based on conversational theory is available [10], for instance. The present formalism generalizes the specification and interpretation of this kind of applications in a systematic and simple way.

In order to apply these intuitions in a practical setting, we have developed a notion of "dialogue model". A dialogue model is defined in terms of conversational situations. A situation is an abstraction of the system's information state defined in terms of the system's expectations, the actions that need to be performed when one among those expectations is satisfied, and the situation that is reached when the corresponding action is performed. For instance, in the *Who is who* scenario of the *Robocup@home* competition[1], a robot may be in a "to be presented" situation in which the participants of the game will be introduced, or in a "searching" situation in which the robot navigates in a room with the purpose of identifying a person.

Expectations may be intentional (i.e. expressed by the interlocutor) or natural (i.e. events that happen in the world). In case the input information cannot be interpreted as an expectation, a recovery dialogue model can be invoked in order to restore the common ground, and "situate" the computational agent in the conversation again. In HCI involving speech, vision and motor behavior, the limitations of recognition and interpretation processes and the presence of noise, make the use of general recovery strategies indispensable to keep the computational agent "grounded" along the interaction.

The specification of expectations and actions is abstract in relation to the modality in which an intention is expressed, and also in relation to the actual concrete expression or input message that conveys the expectation. For instance, in the *Follow me* scenario of *Robocup@home*, a robot may be in a following situation in which one of the expectations may be to be asked to stop (i.e. an action-directive). This expectation may be expressed by a human interlocutor through spoken language, or through a stop gesture presented visually to the robot. Expectations are also specified abstractly in relation to the actual message through which they are communicated. For instance, in the same scenario, the robot may be asked to stop through the expressions "robot, please stop" or simply through "stop". In these two examples the same speech act (direct) is expressed with different lexical and syntactic content, and there is a very large number of forms, containing possible disfluencies and speech repairs, that can be used to express the same speech act. In addition, the same intention can be expressed through indirect speech acts, as in "could you please stop?" where a question is used to express the imperative, or even through "there is someone in front of you", where the command is also an indirect speech act expressed through an assertion with unrelated lexical content.

In the present framework, input information, either linguistic or visual, is taken as evidence to satisfy one among the expectations of the current situation. In this regard, the present model has a strong Bayesian flavor, as the context and expectations codified in advance in dialogue models correspond to *a priori* information used in the

[1] www.robocup2010.org

interpretation, and the concrete input messages provide likelihoods (i.e. evidence given a possible expectation), and altogether, an interpretation given the evidence is the best interpretation hypothesis resulting from weighting the expectations in the current context with the current likelihoods.

In the rest of the paper we present the formalism in more detail. The specification and interpretation of dialogue models is presented in Section 2. In Section 3 we present a cognitive architecture for the definition of intelligent multimodal agents. We focus on the perceptual component of the agent that maps external information into interpretations. In this section we also introduce a notion of *modality*, which involves a perceptual device and a format to encode external information from a particular recognition perspective. In Section 4 we provide references to some applications built within the present framework, and also to work in progress.

2 Representation and Interpretation of Dialogue Models

We define multimodal HCI applications in terms of a set of dialogue models. A dialogue model is an abstract specification of a conversational protocol, and an application domain is characterized as a set of dialogue models. Dialogue models are interpreted directly by an interpreter program, which we call the "dialogue manager". A dialogue model is defined in terms of a set of situations, one of which is designated as the initial, and at least one of them is designated as the final one. Dialogue models have a graphical representation where situations are represented through nodes and situation relations are represented through directed links. Every link has a label of the form α:β, where α stands for an expectation (either intentional or natural) and β stands for the action that is performed by the system when the expectation α is satisfied in the world in the current situation s_i; as a result of performing such action, the system moves to the situation s_j at the end of the link. This is illustrated in Figure 1. The expressive power of the formalism corresponds to recursive transition networks, augmented with functions standing for expectations, actions and next situations (F-RTN), as introduced in [13] and generalized below.

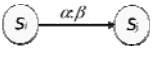


Fig. 1. Graphical Representation of Dialogue Models

The expectation α and the action β may by empty (i.e. ε); in the former case there must be only one output link of situation s_i, and the corresponding action β is performed deterministically; in the latter, the system moves to situation s_j performing no action at all. Expectations can be concrete; for instance, if the system makes an offer, it moves to a situation including two possible concrete expectations: to accept the offer or to reject it. These are expected concrete intentions that can be represented with a constant label or a grounded predicate. Concrete actions, either linguistic or in other modalities, can be represented with these basic representational devices too. For instance, in an application in which a robot guiding a poster session moves from poster *a* to poster *b*, β = "move(*a*, *b*)". The label β stands for complex actions that we

call "rhetorical structures"; these can be composed of a number of basic actions, roughly along the lines of Rhetorical Structure Theory or RST [7]. Instances of rhetorical structures are specified and performed or rendered in any of the system's output modalities. In the present model, β is the specification of a behavior, which can be rendered as text, speech, or as an image (i.e. as an external representation), or performed as a motor action; however, β may also stand for an internal action, like a reasoning or problem-solving task, which is specified and executed when its associated expectation α is satisfied.

The representation of expectations and actions in terms of speech acts with their corresponding parameters permits to recall the interaction history at this intentional level. Once the links of a graph have been travelled, all expressions representing expectations and actions are "grounded". Accordingly, the sequence of situations with the labels of the links travelled in a particular interaction constitute the discourse or anaphoric context of such an interaction, where the predicates are intentional (e.g. assert, order, offer, etc.), and their arguments are the task domain propositions expressed and understood by the system. This information is recalled every time a situation is interpreted by the dialogue manager making this information available for the resolution of further anaphoric references that depend on the discourse context.

In the present version of the formalism, expectations and actions can be specified with predicates including free variables or propositional functions. For instance, in a situation in which the system has asked the user for his or her name, the user is expected to satisfy such an information request with the corresponding assertion. In this case the expectation has the form α = "assert(name(*x*))". Free variables within the body of expectations need to be assigned values from the world through perception, providing an indexical interpretation component to the system. The user may provide his or her name through spoken or written text, and he or she may use different expressions for this purpose, like "Peter", "my name is Peter", "hum… shall I tell you my name? Ok, my name is Peter", etc. However, in all these scenarios the overall purpose of the perceptual component is to assign the value "Peter" to the variable *x*, in order to satisfy the predicate "assert(name(Peter))". We use the term "indexical" in the sense that terms or expressions have values in a particular location and time, and also in relation to an agent (e.g. I, you), this is, in relation to the *indices* of the situation. Actions can also be specified as propositional functions; for instance, a linguistic greeting and acknowledgment can be represented with β = "greet(*x*)". If the user provides his or her name in this situation, and there is an output link labeled with "assert(name(*x*)):greet(*x*)", the system will produce, for instance, "Hi Peter, nice to meet you". In this case the variable is the same, and once its value is assigned by perception in the situation, it is also bound in the corresponding action's description. Situations are also parametric objects. So, the arguments of a situation can be bound to the values taken by the arguments of expectations and actions. This is an additional mechanism that permits the flow of information through the links of the graph along the interaction.

Expectations, actions and also the relations between situations can vary along the course of the interaction. In order to model this kind of variation, previous versions of the formalism permitted the specification of actions through functions; this expressive power has now been extended to model expectations and the following or next situations in terms of explicit functions too, as illustrated in Figure 2. These functions

have as their main argument the interaction's history. Whenever the dialogue manager reaches a situation that has a functional specification, the functions are evaluated first and the links are interpreted using the corresponding functions' values. Also, we have introduced a notation for the functional definition of the next situation. For this, the function *h* itself is represented by a small dot, and its possible values (i.e. its possible next situations) by dashed-links, as illustrated in Figure 2.

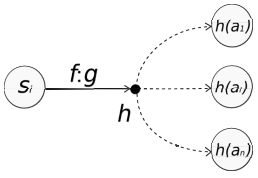


Fig. 2. Functional representation of expectations, actions and transitions

Consider an application in which a robot is the guide of a poster session. A partial specification of the corresponding dialogue model is shown in Figure 3. In this model there is a situation s_{i-1} in which the system makes the offer β_{i-1} = "offer(*a*, *b*)" (where *a* and *b* are the posters that can be explained) and moves to situation s_i. At this situation the system has the expectation that the user accepts the offer and asserts which poster he or she wants to see, so one output link of s_i has the label α_i = "user wants to see *x*, where *x* = *a* or *x* = *b*" (i.e. a propositional function); in the case this expectation is satisfied, the system needs to produce the corresponding explanation, so β_i = "explain(*x*)".

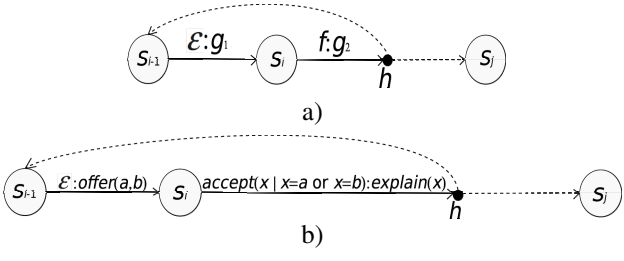


Fig. 3. A functional dialogue model

Suppose now that in an actual interaction the user asserts that he or she wants to see poster *a* at situation s_i; then, the expectation "user wants to see *a*" is satisfied, the action "explain(*a*)" is performed and the system moves to situation s_{i+1}, to continue with the interaction. Now suppose that further on in the same interaction the system reaches again situation s_{i-1}; in this situation the expectations and actions have changed because the user already visited and received the explanation of poster *a*. So, β_{i-1} = "offer(*b*)" and α_i = "user wants to see *x*, where *x* = *b*". Furthermore, suppose that s_{i+1} = s_{i-1}, as after the system has explained a poster, it is reasonable to ask the user if he or she wants to know about another one. This cycle can proceed through the tour until there are no more posters to explain; then, the system goes to a different situation s_j,

to continue with the interaction. Figure 3.a illustrates the functional specification and Figure 3.b the values of the functions at a particular interaction state. The dialogue model can be further elaborated to permit clarifications, and allow the user to see an explanation more than once, if he or she wants to do so.

These functions can also be used to access domain specific and general knowledge of the computational agent. Knowledge at this level is represented in a modality independent fashion, has a propositional character and it is highly independent of perception. As an analogy to the corresponding psychological notion, this kind of knowledge is "stored" in the system's "semantic memory" [15].

Although expectations and actions are specified in a modality independent way, situations have types according to the perceptual modality or modalities through which the input information is recognized and interpreted. The two basic input modalities are *listening* and *seeing*. The formalism also permits the specification of multimodal situations, for instance *listening and seeing* and *listening or seeing*. The conjunctive situation can be used when an intention needs to be expressed through both channels (e.g., for the interpretation of gestures supported by language); the disjunctive one, in turn, can be used when the expectation can be expressed through either of the two channels, as it is the case in the stop situation of the *Follow me* scenario mentioned above. The system can be extended straightforwardly with situation types for other input modalities (e.g. click buttons, laser, sonar, buttons, etc.), or a combination of modalities, depending on the particular input devices of an HCI application.

In addition to the modalities oriented to collect and interpret input information, the specification of dialogue models includes a *recursive* situation, which stands for a full dialogue model. When a situation of this type is reached, the current dialogue model is pushed down into a stack, and the embedded dialogue model is loaded for interpretation at its initial situation. This abstraction capability permits the construction of large applications in a modular and simple way. All identifiers within the body of dialogue models have a local scope, making the interpretation of dialogue models highly decoupled. When a dialogue model reaches a final situation, the dialogue model at the top of the stack, if any, is popped up and its interpretation is resumed at the situation that follows the current recursive one. Each output link of recursive situations has a continuation label (which represents a speech act too) in the place of its corresponding expectation α (i.e. there is a set of intentions that can be expressed or expected events that happen when the recursive situation is interpreted). Similarly, final situations have a unique output link with a continuation label indicating that the dialogue model was terminated in that particular situation and no other (i.e. the embedded dialogue model "produced" a particular expectation). So, the output transition of a recursive situation is determined by the output link whose expectation α matches with the continuation label of the final situation of the embedded dialogue model. Although the recursive situation was available in the original presentation of the formalism, the continuation scheme described here is a new feature introduced in the current version. We refer to this generalized computational mechanism as Functional Recursive Transition Networks or F-RTN.

3 A Cognitive Architecture for Multimodal HCI

In this section we describe an architecture to assign interpretations to input messages in one or more modalities in terms of the agent's expectations in a situation, the context represented by dialogue models, and also in terms of memory. This architecture is also used to specify and perform or render behaviors, also in one or more modalities, that result from the satisfaction of the system's expectations. The architecture has three conceptual levels that we call representation/inference, interpretation/specification and recognition/rendering, in addition to the semantic and episodic memory, as shown in Figure 4.

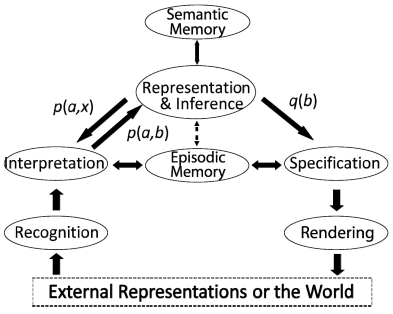


Fig. 4. Cognitive architecture

The present cognitive architecture is interaction-oriented. There is a main interaction cycle involving recognition -> interpretation -> representation -> inference -> specification of behavior -> rendering. In addition, the top-level level of representation and inference is also interaction-oriented, as it corresponds to the dialogue models with its graph-oriented computation (i.e. F-RTN). The system's semantic memory can be accessed through the functions representing expectations, actions and also next situations, as mentioned above.

Recognition and interpretation correspond to low and high-level perception. Recognition is mainly a bottom-up process that translates the input information into an image or pattern in a particular coding format, involving a recognition perspective (i.e. a particular recognition algorithm), mostly in a context independent manner. However, no meaning or interpretation is assigned to the product of this recognition level. These patterns are modality specific "images" that correspond to external representations (e.g. messages) or to the world, as presented to the agent through a particular recognition device. We refer to this kind of patterns as "uninterpreted images". Human speech is recognized by an ASR system to produce a text in a particular format (e.g. ASCII), but the basic functionally of the ASRs does not assign an interpretation to such a text, and the text as such is an uninterpreted image. Someone with knowledge of the Greek alphabet but no knowledge of Greek may be aware that a piece of paper with Greek symbols on it is indeed a text (i.e. a coding scheme), although he or she would not be able to know what it means. On the vision case, an external image can be processed through the SIFT algorithm [6], and the image is codified in this case as a set of vectors of a large dimensionality, where each

vector represents a salient point of the visual image that is invariant to translation and rotation. In this scheme, the set of vectors codify the image but, as in the case of text, the pattern is an uninterpreted image. This SIFT algorithm has been used in our framework for the construction of applications and demos with vision capabilities [1]. Furthermore, the same visual image can be processed by an alternative vision recognition algorithm and produce a different pattern in a different coding format, from a different recognition perspective (e.g. codified through a set of features, like edges and corners); we have used this dual interpretation of visual images for the interpretation of gestures and gesture referents [3, 4].

A particular implementation of the present architecture has a finite number of coding formats of this kind. Each coding format is associated with a particular recognition device, and also to a specific algorithm. In our terminology, each of these coding formats corresponds to a "modality". The set of coding schemes or modalities characterizes to the kinds of images that an agent can perceive from the world through its recognition devices and store in its memory. On the output channel, the rendering devices may transform images into external representations (e.g. synthesizing speech or rendering a picture on the screen), or produce the motor actions of the agent on the world directly.

We turn now to the middle interpretation layer of the architecture. This corresponds to high-level perception and its purpose is to assign interpretations to uninterpreted images at specific dialogue model situations. The central component of this level is an interpreter program (which is different from the dialogue model's interpreter). Its inputs are the expectations of the agent at the interpretation situation (top-down) and one or more recognized uninterpreted images, of one or more modalities (bottom-up). Accordingly, the interpretation of input messages or information is a heterarchic process. A particular implementation of the architecture has an interpreter at this level for each modality or combination of modalities useful in the application domain. Also, a situation type is defined in the dialogue models for each of these interpreters.

The outputs of the interpreters at this middle level are "interpretations". Although the interlocutor expresses concrete messages, the agent "understands" speech acts (which ought to correspond to the interlocutor's intentions expressed through the messages). Natural events that happen in the world (but are expected) are interpreted in the same intentional fashion. The interpretation process is also illustrated in Figure 4. There, the propositional function $p(a, x)$ is an expectation (a propositional function expressing a limited abstraction), which is satisfied by the message or event recognized from the world, and produces the interpretation $p(a, b)$. In this example, the binding of the variable x with the value b determines the action $q(b)$ which is also further specified and performed by the output channel.

We turn now to the episodic memory module. This memory is constituted by a set of associations between an interpretation (or meaning) and an uninterpreted image (or clusters of uniterpreted images). These associations can be used for interpretation and recall. In the former case, recognition produces an uninterpreted image, which becomes the index of its associated interpretation; conversely, an interpretation (or meaning) can render its associated uninterpreted image. Meaning and uniterpreted images can be complex structured objects, but in the simplest case meanings correspond to individual concepts and uninterpreted images correspond to "images"

of the corresponding individual in the modalities of the system. At this stage of our work, the memory units available to a particular application are collected and stored beforehand, using the same recognition devices and modalities that are available in the application. In order to make a productive use of these memories, expectations are used as indices to their corresponding memory units, and only the images indexed by the expectations in a particular interpretation situation may be retrieved to assign an interpretation to the image recognized in the situation. The present memory module loosely resembles some of the properties of long term "episodic memory" [15], and we adopt this terminology for this reason.

Also, the matching process is qualitative and approximate. In the case of text, for instance, an individual concept can be represented through a regular expression containing a large number of forms that can express such a concept. In the visual case, a concept is associated with a view or sets of views stored in the SIFT format. Such a concept can be visually retrieved if the geometric distance from the image recognized in a particular situation is near enough to one of its corresponding images in memory [1]. Finally, uninterpreted images could also be accessed through the functions specified in the dialogue models, as illustrated with a dashed-line in Figure 4. This functionality is aimed to support heterogeneous reasoning and problem-solving, like diagrammatic reasoning [11].

4 Current Results and Further Work

We have applied the present framework for the construction of several applications, including a conversational robot capable to guide a poster session through spoken Spanish with the support of other output visual modalities[2]. We have also developed a system able to play the game "Guess the card" with members of the general public through spoken Spanish and vision, at the permanent exhibition of UNAM's science museum Universum [8, 9]. At the moment we are also developing a new version of the guide tour demo, this time including the capability of interpret pointing gestures [3, 4]. We are also working in the integration of acoustic scene analysis to allow a robot to locate the location of its human interlocutor through his or her voice, and also to filter diverse sound sources and noise to improve the ASR system's performance [14]. The framework seems to be general enough to be used in a wide range of domains and applications.

References

1. Aguilar, W., Pineda, L.A.: Integrating Graph-Based Vision Perception to Spoken Conversation in Human-Robot Interaction. In: Cabestany, J., et al. (eds.) IWANN 2009. LNCS, vol. 5517, pp. 789–796. Springer, Heidelberg (2009)
2. Allen, J., Core, M.: Draft of DAMSL: Dialog Act Markup in Several Layers Annotation Scheme. Department of Computer Science, Rochester University (1997)
3. Avilés, H., Meza, I., Aguilar, W., Pineda, L.A.: Integrating Pointing Gestures into Spanish-Spoken Dialog System for Conversational Service Robots. In: Proceedings of ICAART 2010, Valencia, España, pp. 585–588 (DVD) (2010)

[2] Videos of demos are available at `http://leibniz.iimas.unam.mx/~luis/`

4. Avilés, H., Alvarado-González, M., Venegas, E., Rascón, C., Meza, I., Pineda, L.A.: Development of a Tour-Guide Robot Using Dialogue Models and a Cognitive Architecture. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 512–521. Springer, Heidelberg (2010)
5. Clark, H., Schaefer, E.: Contributing to Discourse. Cognitive Science 13, 259–294
6. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004) (1989)
7. Mann, W.C., Thompson, S.: Rhetorical Structure Theory: Towards a functional theory of text organization. Text 8(3), 243–281 (1988)
8. Meza, I., Pérez-Pavón, P., Salinas, L., Avilés, H., Pineda, L.A.: A Multimodal Dialogue System for Playing the Game "Guess the Card". Procesamiento del Lenguaje Natural (44), 131–138 (2010)
9. Meza, I., Salinas, L., Venegas, E., Castellanos, H., Chavarria, A., Pineda, L.A.: Specification and Evaluation of a Spanish Conversational System Using Dialogue Models. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 346–355. Springer, Heidelberg (2010)
10. Nakauchi, Y., Naphattalung, P., Takahashi, T., Matsubara, T., Kashiwagi, E.: Proposal and Evaluation of Nat. Lang. Human-Robot Interface System based on Conversation Theory. In: Proceedings of the 2003 IEEE Int. Conf. on Robotics and Automation, Taipei, Taiwan, September 14-19, pp. 380–385 (2003)
11. Pineda, L.A.: Conservation principles and action schemes in the synthesis of geometric concepts. Artificial Intelligence 171(4), 197–238 (2007)
12. Pineda, L.A., Estrada, V., Coria, S., Allen, J.: The Obligations and Common Ground Structure of Practical Dialogues, Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial 11(36), 9–17 (2007)
13. Pineda, L.A.: Specification and Interpretation of Multimodal Dialogue Models for Human-Robot Interaction. In: Sidorov, G. (ed.) Artificial Intelligence for Humans: Service Robots and Social Modeling, SMIA, México, pp. 33–50 (2008)
14. Rascón, C., Avilés, H., Pineda, L.A.: Robotic Orientation towards Speaker in Human-Robot Interaction. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 10–19. Springer, Heidelberg (2010)
15. Tulving, E.: Memory systems: episodic and semantic memory. In: Tulving, E., Donaldson, W. (eds.) Organization of Memory, pp. 381–403. Academic Press, New York (1972)

Optimization of Investment Options Using SQL

Jose Torres-Jimenez[1], Himer Avila-George[2],
Nelson Rangel-Valdez[1], and Jorge Martinez-Pena[1]

[1] CINVESTAV-Tamaulipas, Information Technology Laboratory
Km. 6 Carretera Victoria-Monterrey, 87276 Victoria Tamps., Mexico
[2] Universidad Politécnica de Valencia - Instituto de Instrumentación para Imagen Molecular (I3M),
Camino de Vera s/n, 46022 Valencia, Spain
jtj@cinvestav.mx, hiavgeo@posgrado.upv.es, nrangel@tamps.cinvestav.mx, jmartinez@tamps.cinvestav.mx

Abstract. This paper presents a novel use of SQL language to solve a practical optimization problem to find the portfolio size and the quantity of money for securities. This problem is known as the Portfolio Selection Problem (PSP). The approach was tested on 9 random instances of PSP. Each instance has up to 12 securities and 50 different choices of money. Each security follows a non-linear profit model. The limitations of our approach are bounded by the computer resources, given that potentially SQL constructs the Cartesian product of the investment options, but it has the advantages of not requiring complex structures and it is easy to understand.

Keywords: Optimization, Portfolio Selection Problem, SQL query.

1 Introduction

The portfolio selection problem is the activity involved in selecting a portfolio of stocks that meets or exceeds an investor's stated objectives [9]. This process is fundamentally based on two variables, expected return and risk. In portfolio selection, the most popular risk definition is variance which, according to Markowitz in [14], means that the greater deviation from the expected value, the less likely the investor can obtain the expected return. Markowitz established that investors would optimally hold a mean-variance efficient portfolio [2]. This is a portfolio with the highest expected return for a given level of variance. Markowitz assumed that investors are risk averse, this means that they accept higher risk only if they get higher expected return. Therefore, investors will prefer a portfolio that offers at least the same expected return than a single stock but with an overall lower risk. This is named diversification. Unsystematic risk is generated by the performance of the companies or industries; therefore, a good selection of the companies is important for the performance of a selected portfolio.

The first attempt to solve the PSP problem was made through the *mean-variance* model shown in [14]. In this model the minimization of the risk was done

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 30–39, 2010.

through the minimization of the variance. After this work, the PSP problem has been solved through the establishment of constraints over the original model. In [11] a linear version of the original objective function of the PSP problem shown in [14] was proposed. More complex objective functions based on *skewness* and *semi-variance* are studied in [10] and [15], respectively. In [3,16,17] local search techniques for solving the PSP problem are proposed. More recent approaches that solves the PSP problem are based on metaheuristics [6] and web-based decision support systems [7].

In general, PSP has been solved mostly through the use of procedural languages like the C language. The Structured Query Language (SQL) is a language that is as powerful as the relational calculus [4] and the relational algebra [5] in the management of stored data. SQL is easy to use and allows the users to express the desired results of a query in a high-level data language. Some work found in the literature, as the one presented at [12], have made use of a declarative language like SQL to deal with constrained optimization. However, to the best of our knowledge, there is no approach reported in the literature that uses SQL to solve PSP.

In this paper we propose a model to solve instances of the PSP problem using a well known non-procedural data access sublanguage namely SQL. The rest of the paper is organized as follows: section 2 presents the description of the PSP problem. Section 3 describes the optimization model based on SQL that solves the PSP problem. Section 4 presents the experimental design used to test the proposed solution. Section 5 presents the conclusions derived from our research.

2 Portfolio Selection Problem (PSP)

In this section we give a formal definition for the PSP problem. After that, we show the solution of an instance of the PSP problem. Finally, we end the section analyzing the search space of the problem.

2.1 Formal Definition of the Portfolio Selection Problem

According to [14] the process of selecting an investment portfolio would be divided in two stages. The first one consists in finding the behavior of securities in the future according with experiences and observations. The second stage takes the most relevant securities and transforms them into an investment portfolio. The investment portfolio involves those securities that minimize the risk and maximize the profit of the investment.

Assuming that the maximum risk has been established by the investor (this implies that the securities were already selected according to the defined risk), this paper presents the PSP as an optimization problem whose objective function is to maximize the expected return. The solution of the problem is subject to a set of securities and a quantity $\mathcal{Q}$ to be invested. The size of the optimum investment portfolio, the selected securities in which to invest and the quantity to invest in each security will be delivered at the end of the solution process.

The PSP problem can be formally defined as follows: given a set of possible quantities of money to be invested $\mathcal{I} = \{q_1, q_2, ...\}$, a set of securities $S = \{s_1, s_2, ...\}$ and a matrix β of size $N \times K$, where $N = |\mathcal{I}|$ and $K = |\mathcal{S}| + 1$, that indicates the profit of investing a quantity of money in a security; which is the vector e of size $|\mathcal{S}|$ that maximizes the profit satisfying the requirement that we expend exactly a prespecified quantity of money? Formally this definition is shown in Equation 1:

$$\begin{aligned} &maximize \quad \sum_{j=1}^{K-1} \beta_{e_j,j} \\ &subject\ to: \\ &\qquad \sum_{j=1}^{K-1} \beta_{e_j,1} = \mathcal{Q} \\ &\qquad e_j = i | q_i \in \mathcal{I} \end{aligned} \tag{1}$$

The matrix β is a matrix of investments and profits. The first column of this matrix contains the different choices of money to be invested i.e., the set of investments $\mathcal{I}$; the quantities are in increasing order starting from 0. The rest of the columns are the set of securities S where money can be invested. The profit of investing the quantity $q_i \in \mathcal{I}$ on security $s_j \in S$ is found in element $\beta_{i,j+1} \in \beta$. The value for each element $e_j \in e$ is the index i of the quantity $q_i \in \mathcal{I}$ that must be invested in security $s_j \in S$ such that Equation 1 is maximized. The quantity q_1 of the first row of the matrix β must be 0 (this enables the possibility that a security does not belong to a portfolio).

The set of the investments $\mathcal{I}$ is constructed with a maximum quantity $\mathcal{Q}$ to invest. The quantity is divided in N different choices of money, which will form the set $\mathcal{I}$. The first quantity $q_1 \in \mathcal{I}$ will be always 0. The quantity q_i, for $2 \leq i \leq N$ will be defined according to Equation 2.

$$q_i = (i-1) * \frac{\mathcal{Q}}{N-1} \tag{2}$$

Summarizing, the matrix $\beta_{N \times K}$ contains the values of the different $N + 1$ investments $\mathcal{I}$ at the first column and the profit, per different choice of money, for each of the $K - 1$ securities in the rest of the columns.

2.2 A Portfolio Selection Problem Instance

Following the definition already given, an instance for the PSP problem is defined with a matrix $\beta_{N \times K}$ and a maximum quantity $\mathcal{Q}$. Table 1 shows an instance of the PSP problem. The maximum quantity to invest is $\mathcal{Q} = 1000$. There are $N = 11$ different choices of money ($\mathcal{I} = \{$ 0, 100, 200, 300, 400, 500, 600, 700, 700, 900, 1000 $\}$) and $K = 4$ securities ($S = \{s_1, s_2, s_3, s_4\}$). The profits are defined for each pair (q_i, s_j). Note the existence of a row of profits containing only zeros as values (this allows to have portfolios of different sizes, given that a security could be not included in a portfolio).

Table 1. Instance of the PSP problem represented as a table $\beta_{11\times5}$. The maximum quantity $\mathcal{Q}$ to be invested is 1000.

$\mathcal{I}$	s_1	s_2	s_3	s_4
0	0	0	0	0
100	28	25	15	20
200	45	41	25	33
300	65	55	40	42
400	78	65	50	48
500	90	75	62	53
600	102	80	73	56
700	113	85	82	58
900	132	90	96	60
1000	138	90	100	60

A SQL query can be used to solve the instance shown in Table 1. This query is shown in Table 2. The *FROM* clause enables to work with all the combinations of securities and quantities to be invested, i.e. all the possible portfolios. The *WHERE* clause filters the portfolios allowing only those ones in which the sum of the quantity of money invested equals to the amount of money $\mathcal{Q}$. The *ORDER BY* clause is used because when the results of the query are arranged in descending order of the portfolio profit, the solution with the maximum profit for the instance is found in the first tuple. The investments per security in the best portfolio and the profit are specified by the *SELECT* clause. The profit is formed by the sum of the individual investments of each security in the selected portfolio.

Table 2. A SQL query that solves the Portfolio Selection Problem instance shown in Table 1. The matrix β is taken as a SQL table with fields $\mathcal{I}$, s_1, s_2, s_3, s_4.

SELECT	$\beta_1.\mathcal{I}$,
	$\beta_2.\mathcal{I}$,
	$\beta_3.\mathcal{I}$,
	$\beta_4.\mathcal{I}$,
	$(\beta_1.s_1 + \beta_2.s_2 + \beta_3.s_3 + \beta_4.s_4)$
FROM	β as β_1,
	β as β_2,
	β as β_3,
	β as β_4
WHERE	$(\beta_1.\mathcal{I} + \beta_2.\mathcal{I} + \beta_3.\mathcal{I} + \beta_4.\mathcal{I} = 1000)$

ORDER BY 5 **DESC LIMIT** 1

The solution of the SQL query presented in Table 2 is shown in the Table 3. The best portfolio is of size 4 given that all the securities participate in the solution with an investment greater than 0. The profit of the portfolio is 181.

Table 3. Vector e representing the solution for the PSP instance shown in Table 1

	Security				
	A	**B**	**C**	**D**	**Profit**
$\mathbf{e} \leftarrow$	400	300	100	200	181

2.3 Complexity Analysis of the Portfolio Selection Problem

For a PSP instance, the search space in the SQL query is defined by the set of all the possible portfolios that can be formed given the securities and the choices of money. This set is formed in the clause *FROM* through the cartesian product of the $K-1$ tables involved on it (one per security). All the tables in the *FROM* clause have the same size $N \times K$, as a consequence the size of the search space is N^{K-1}.

Other approaches in the literature [6] solves the PSP problem in two stages. In the first one they find the size of the portfolio and which securities belong to it. In the second stage they find the amount of money to be invested on each security of the portfolio s.t. the expected return is maximized. Given that each security can either be in the portfolio or not, the search space of the first stage is 2^{K-1} (the number of possible combinations). The complexity of the second stage is $(N-1)^i$, because there are $N-1$ choices of money (zero is no longer available) and i is the portfolio size. Given these complexities, the search space of this approach will be completely described by Equation 3.

$$\sum_{i=0}^{k-1} \binom{k-1}{i} (N-1)^i \tag{3}$$

A simple inspection in Equation 3 will reveal that it is equivalent to the search space of the SQL query proposed in this paper. It is important to note that while some approaches first determine the size of the portfolio and the elements that belong to it, and after that they solve the problem of determining the amount of money that must be invested per security, the SQL query approach solves both problems simultaneously keeping the search space unchanged.

The next section presents the generalization of the SQL query that solves a particular instance of the PSP problem.

3 SQL Approach for Solving the Portfolio Selection Problem

Given an instance of the PSP Problem as a matrix β and a quantity $\mathcal{Q}$, we propose an exact approach that finds the investment portfolio e that maximizes the objective function of Equation 1. The approach uses a query in the Structured Query Language (SQL). Once that the query has been created, it is executed in a database management system.

The query shown in Table 2 presents the SQL query required to solve the particular PSP instance shown in Table 1. Table 4 shows the generalization of

that query for any PSP instance. The query is given using the BNF notation [8]. Given the matrix β as a database table with fields $\mathcal{I}$ and $s_i \in S$, the *FROM* clause contains the cartesian product of as much β tables as the number of securities $K-1$ of the instance that is going to be solved using SQL. The *SELECT* clause will select the field representing $\mathcal{I}$ from each table in the clause *FROM*. In addition, the *SELECT* clause includes an extra field that represents the summation of the profits of the fields in the previous filter. The *WHERE* clause will contain the total of the investments of the previously selected fields; this summation equals the amount of money to be invested $\mathcal{Q}$. Finally, the SQL query will sort the results according to the extra field representing the summation of the profits. The descending order given to the results allows to identify the solution as the first tuple of the query.

Table 4. BNF format of the general SQL query that solves the PSP problem

SELECT <security investments>,	
<profit>	
FROM <vector e definition>	
WHERE <constraint definition>	
ORDER BY <number of securities plus 1> **DESC LIMIT** 1	
<security investments>	::= $\beta_1.\mathcal{I}, \ldots, \beta_{K-1}.\mathcal{I}$
<profit>	::= $(\beta_1.s_1 + \beta_2.s_2 + \ldots + \beta_{K-1}.s_{K-1})$
<vector e definition>	::= β as β_1, β as β_2, ..., β as β_{K-1}
<constraint definition>	::= $(\beta_1.\mathcal{I} + \beta_2.\mathcal{I} + \ldots + \beta_{K-1}.\mathcal{I} \leq \mathcal{Q})$
<number of securities plus 1>	::= K

The next section presents the experimental results of using the SQL approach proposed in this paper applied to the solution of random instances of the PSP problem.

4 Experimental Design

In this section we present the experimentation done to test the model based on a SQL query to solve the PSP problem. In order to test the performance of the query we solved 9 random instances of the PSP problem. The generator of the instances is described in the next subsection.

4.1 Random Instances Generator

Following the definition shown in Section 2, a PSP problem instance can be defined through a matrix $\beta_{N \times K}$ and a quantity $\mathcal{Q}$ representing the amount of money to be invested. This section presents the algorithm used to generate random instances of the PSP problem following that definition.

Basically, given the values for N, K and $\mathcal{Q}$, the algorithm establishes the different choices of money and randomly generates the profit for each security.

The different investments $q_i \in \mathcal{I}$ are saved in the first column of matrix β, in positions $\beta_{i,1}$, for all $1 \leq i \leq N$; their values range from 0 to $\mathcal{Q}$ with increments of $\frac{\mathcal{Q}}{N}$.

The profit for each security is determined in two steps. The first step consists in randomly selecting a model for the profit. The second step will propagate the effect of the chosen model over the different choices of money to be invested in a security.

In this approach we proposed four simple non-linear profit models to evaluate the performance of the SQL query when solving random PSP instances. These models uses non-linear functions (natural logarithm $ln(v)$, squared natural logarithm $ln(v)^2$, square root $\sqrt{v}$ and cubic root $\sqrt[3]{v}$) of the different investments q_i, for $1 \leq i \leq N$. Also, these models can involve one or two random values a, b between 0 and 1. Given a security s_j and an investment q_i, whose value is in the position $\beta_{i,1}$ of the matrix β, the profit for the element $\beta_{i,j} \in \beta$ is modified according with the chosen model.

The first model is $\beta_{i,j} = a \cdot \log \beta_{i,1}$. In this model the profit is a fraction of the natural logarithm of a choice of money, a is a random constant.

The second model $\beta_{i,j} = a \cdot (\log \beta_{i,1})^2$ represents the profit as a fraction of the square of the natural logarithm of a choice of money, a is a random constant.

The third and fourth profit models are $\beta_{i,j} = \sqrt{a \cdot \beta_{i,1}}$ and $\beta_{i,j} = \beta \cdot \sqrt[3]{a \cdot \beta_{i,1}}$, respectively. These models are also based in fractions of a choice of money but they defined using square and cubic roots, respectively, a is a random constant.

The profit for all the securities is 0 when the amount of money to be invested is 0, i.e., in the first row of matrix β is all zeros.

Algorithm 4.1 shows the pseudocode of the generator of random matrices. This algorithm output the matrix β representing the PSP instance. Each element $\beta_{i,j} \in \beta$ is defined by this algorithm. Sentence 1 determine the different choices of money for the investment $\mathcal{Q}$. Sentence 2 randomly choose one of the two profit models described in this section. Sentence 3 contains the loop that propagates the effect of the profit model over the current security.

Algorithm 4.1: randomInstanceGenerator$(N, K, \mathcal{Q})$

comment: I*nput* : *N Number of different choices of money*
comment: I*nput* : *K Number of securities minus* 1
comment: I*nput* : $\mathcal{Q}$ *amount of the investment*
comment: O*utput* : β *matrix of PSP instance*
$\Delta \leftarrow \frac{\mathcal{Q}}{N}$
comment: Establish the different choices of money for the investment
for $i \leftarrow 1$ **to** N (1)
do { **if** $i = 1$ **then** $\{\beta_{i,1} \leftarrow 0$ **else** $\{\beta_{i,1} \leftarrow \beta_{i-1,1} + \Delta$
comment: Determine the profit for each security
for $j \leftarrow 2$ **to** K
do { $profitModel \leftarrow$ randomlyChooseProfitModel() (2)
for $i \leftarrow 1$ **to** N (3)
do { $a \leftarrow$ Random$(0, 1)$
$\beta_{i,j} \leftarrow$ ProfitValue$(profitModel, a, \beta_{i,1})$

Table 5. PSP instance randomly generated using Algorithm 4.1

Choice of Money	Security A	B	C	D	E	F	G	H	I	J	K
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
83.33	1.31	1.77	4.10	2.95	10.23	2.22	0.16	19.23	11.91	2.20	5.55
166.67	1.64	2.04	4.75	3.95	13.69	3.14	0.20	25.73	15.94	2.77	7.85
250.00	1.88	2.20	5.12	4.60	15.94	3.84	0.23	29.97	18.57	3.17	9.62
333.33	2.07	2.32	5.39	5.10	17.65	4.44	0.25	33.17	20.55	3.48	11.11
416.67	2.23	2.41	5.60	5.50	19.03	4.96	0.27	35.77	22.16	3.75	12.42
500.00	2.37	2.48	5.77	5.83	20.20	5.43	0.28	37.97	23.52	3.99	13.60

A PSP instance using the algorithm previously described is shown in Table 5. This instance has an investment of $\mathcal{Q} = 100$. There are $N = 7$ different choices of money for the investment (see column 1). The number of securities is $K = 11$.

Table 6 gives the solution for the PSP instance shown in Table 5. This solution is represented by the vector e of size $K - 1$. The elements e_i, for $1 \leq i \leq K - 1$ represent the choice of money for each security $s_i \in S$. The element e_K represents the profit of the portfolio selected. The best profit was 57.45. It was achieved by the portfolio of size 4 formed by the securities **E, H, I, K**. The best investment for each security in the portfolio was $83.33, 166.67, 166.67, 83.33$, respectively. As you can see, the SQL query solved both the best portfolio size and profit.

Table 6. Vector e representing the solution for the PSP instance shown in Table 5

	Security A	B	C	D	E	F	G	H	I	J	K	Total Profit
e ←	{0,	0,	0,	0,	83.33,	0,	0,	166.66,	166.66,	0,	83.33}	57.45

The next section presents the experimental results obtained of solving random instances generated using the algorithm presented in this section, with the SQL query described in Section 3.

4.2 Computational Experiments

The random instances generator was implemented in C language and compiled with gcc. The instances were generated in a Desktop Computer with an Intel(R) Core(TM) 2 CPU 2400 6400 @ 2.13 Ghz processor, 3Gb of RAM and Ubuntu 8.10 Intrepid Ibex Operating System.

Table 7 shows the 10 random instances of the PSP problem generated using the algorithm described in Section 4.1. The column 1 presents the instances. The column 2 shows the number of different choices of money to be invested. The number appearing in column 3 represents the number of securities involve in the PSP instance (including an extra column to describe the different choices of money). The column 4 contains the amount of money to be invested.

Table 7 also shows the results from solving each PSP instance through a SQL query, based on the one described in Section 2. The query for each instance was

Table 7. PSP instances and their solution

Instance	N	K	$\mathcal{Q}$	Portfolio	Profit	Time (sec.)
1	5	13	100	{0, 0, 0, 20, 0, 0, 20, 20, 0, 0, 40, 0 }	20.62	4497.55
2	6	12	500	{0, 0, 0, 0, 83.33, 0, 0, 166.66, 166.66, 0, 83.33 }	57.45	3588.78
3	8	11	1000	{125, 125, 125, 0, 250, 0, 125, 125, 0, 125}	40.96	5465.39
4	10	10	100	{30, 10, 10, 10, 10, 10, 20, 0, 0}	20.64	3158.89
5	13	9	500	{153.84, 0, 76.92, 0, 38.46, 153.84, 0, 76.92}	37.026	1734.03
6	15	8	1000	{200, 133.33, 400, 0, 66.66, 133.33, 66.66}	43.806	261.99
7	25	7	100	{4, 12, 40, 8, 0, 36}	8.605	231.09
8	30	6	500	{16.66, 200, 16.66, 33.33, 233.33}	29.82	17.37
9	50	5	1000	{260, 720, 0, 20}	37.019	2.86

executed in MySQL 5.0.67. Table 7 shows the portfolio with the best profit for each instance in column 5. The portfolio is represented as the amount of money to be invested in each security, the securities are arranged from 1 to K - 1. The profit resulting from the portfolio is shown in column 6. Column 7 shows the time taken by MySQL to solve the SQL query.

The time spent in the solution of the PSP problem instances grows considerably when the number of columns and/or the number of rows of the matrix increases. This behavior was expected given that the PSP problem belongs to the class of NP-complete problems[1,13], the search space grows exponentially given that the SQL query is an exhaustive approach that may require to analyze all the potential solutions to give the answer wished. Given that the purpose of the algorithm was to show how a hard optimization problem can be easily solved through a simple sentence of a declarative language such as SQL, and not for the solely purpose of optimization, the algorithm was not compared against other existing approaches.

5 Conclusions

This paper presents a novel model for solving the PSP problem using a SQL query. The proposed SQL query simultaneously allows to find the size of the portfolio, the elements of the portfolio and the amount of money per security required to maximize the expected return of a given PSP problem instance.

The search space of the model proposed in this paper to solve the PSP problem is equivalent to the search space of solving the same problem using two steps, first solving the size of the portfolio and then finding its element and after that finding the best combination of investment per security.

The size of the PSP instances that can be solved using a SQL query is bounded by the database management system capabilities. However, the use of SQL to solve PSP avoids the use of complex structures, like the ones used by procedural languages. This makes this approach easy to use and understand.

Currently we are trying to solve large instances of the PSP problem using the approach presented in this paper.

Acknowledgements. 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.

References

1. Bienstock, D.: Computational study of a family of mixed-integer quadratic programming problems. Mathematical Programming: Series A and B 74(2), 121–140 (1996)
2. Campbell, J., et al.: The Econometrics of Financial Markets. Princeton University Press, Princeton (1997)
3. Chang, T.J., Meade, N., Beasley, J.E., Sharaiha, Y.M., Stanley, M., Witter, D.: Heuristics for Cardinality Constrained Portfolio Optimisation. Computers and Operations Research 27, 1271–1302 (1998)
4. Codd, E.F.: A relational model of data for large shared data banks. ACM Commun. 13(6), 377–387 (1970)
5. Codd, E.F.: Further Normalization of the Data Base Relational Model. IBM Research Report, San Jose, California, FJ909 (1971)
6. Coutino-Gomez, C.A., Torres-Jimenez, J., Villarreal-Antelo, B.M.: Heuristic Methods for Portfolio Selection at the Mexican Stock Exchange. In: Liu, J., Cheung, Y.-m., Yin, H. (eds.) IDEAL 2003. LNCS, vol. 2690, pp. 919–923. Springer, Heidelberg (2003)
7. Dong, J., Du, H.S., Wang, S., Chen, K., Deng, X.: A framework of Web-based Decision Support Systems for portfolio selection with OLAP and PVM. Decision Support Systems 37(3), 367–376 (2004)
8. Friedman, J.: A computer system for transformational grammar. ACM Commun. 12(6), 341–348 (1969)
9. Ghasemzadeh, G., Archer, N.P.: Project portfolio selection through decision support. Decision Support Systems 29(1), 73–88 (2000)
10. Hiroshi, K., Ken-ichi, S.: A mean variance-skewness portfolio optimization model. Journal of the Operations Research Society of Japan 2(37), 173–187 (1995)
11. Konno, H., Yamazaki, H.: Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. Management Science 37(5), 519–531 (1991)
12. Mancini, T., Flener, P., Hossein Monshi, A., Pearson, J.: Constrained optimisation over massive databases. In: Proceedings of RCRA 2009, the 16th RCRA International Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion, vol. 589 (2009)
13. Mansini, R., Speranza, M.G.: Heuristic algorithms for the portfolio selection problem with minimum transaction lots. European Journal of Operational Research 114(2), 219–233 (1999)
14. Markowitz, H.: Portfolio Selection. The Journal of Finance 7(1), 77–91 (1952)
15. Markowitz, H., Todd, P., Xu, G., Yamane, Y.: Computation of mean-semivariance efficient sets by the Critical Line Algorithm. Annals of Operations Research 45(1-4), 307–317 (1993)
16. Rolland, E.: A tabu search method for constrained real-number search: Applications to portfolio selection. Technical report, Department of Accounting and Management Information Systems, Ohio State University, Columbus (1997)
17. Schaerf, A.: Local Search Techniques for Constrained Portfolio Selection Problems. CoRR, cs.CE/0104017 (2001)

A Computational Method for Defeasible Argumentation Based on a Recursive Warrant Semantics

Teresa Alsinet[1], Ramón Béjar[1], and Lluís Godo[2]

[1] Department of Computer Science – University of Lleida
C/Jaume II, 69 – 25001 Lleida, Spain
{tracy,ramon}@diei.udl.cat
[2] Artificial Intelligence Research Institute (IIIA-CSIC)
Campus UAB - 08193 Bellaterra, Barcelona, Spain
godo@iiia.csic.es

Abstract. In a recent paper [2] the authors have formalized a recursive semantics for warranted conclusions in a general defeasible argumentation framework based on a propositional logic. The warrant recursive semantics is based on a general notion of collective (non-binary) conflict among arguments allowing to ensure direct and indirect consistency properties. This general framework has also been extended with levels of defeasibility and with a level-wise recursive definition of warranted and blocked conclusions. In this paper we focus on the recursive semantics for the particular framework of Defeasible Logic Programming (DeLP) extended with levels of defeasibility, called RP-DeLP, for which we characterize programs with a unique output (extension) for warranted conclusions, and we design, for this type of programs, an algorithm for computing warranted conclusions in polynomial space and with an upper bound on complexity equal to P^{NP}.

Keywords: defeasible argumentation, recursive semantics, computational aspects.

1 Introduction and Motivation

Possibilistic Defeasible Logic Programming (P-DeLP) [1] is a rule-based argumentation framework which incorporates the treatment of possibilistic uncertainty at the object-language level. Indeed, P-DeLP is an extension of Defeasible Logic Programming (DeLP) [17] in which defeasible rules are attached with necessity degrees (belonging to the real unit interval $[0, 1]$) expressing their belief strength. In P-DeLP, warranted (justified) conclusions or beliefs with a maximum necessity degree are formalized in terms of an exhaustive dialectical analysis of all possible arguments.

Because the dialectical tree-based P-DeLP semantics for warranted conclusions does not satisfy the *indirect consistency* property[1], in [3,2] a new recursive semantics ensuring that property has been investigated. In particular, following the ideas of an approach defined by Pollock [19], the authors have formalized in [2] a recursive semantics for

[1] This property is satisfied when the set of warranted conclusions is consistent with respect to the set of strict (non-defeasible) rules of a program and was identified in [6] as a basic postulate that rule-based argumentation systems should satisfy.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 40–49, 2010.

warrant in a general defeasible argumentation framework based on a notion of collective (non-binary) conflict among arguments. This general framework has also been extended by introducing levels of defeasibility in the knowledge base and providing a level-wise recursive definition of warranted and blocked conclusions. In this setting one can distinguish between warranted and blocked conclusions. A warranted conclusion is a ultimately justified conclusion which is based only on warranted information and which does not generate any conflict, while a blocked conclusion is a conclusion which, like warranted conclusions, is based only on warranted information but which generates a conflict. Finally, in the same paper, this recursive warrant semantics is particularized to the framework of P-DeLP, the resulting formalism being called RP-DeLP (Recursive P-DeLP), and circular definitions of conflict between arguments that lead to multiple extensions (outputs) are identified by means of *warrant dependency graphs*.

In this paper, after overviewing in Section 2 the main elements of the warrant recursive semantics for RP-DeLP and the unique output property for RP-DeLP programs, in Section 3 we design for this type of programs an algorithm for computing warranted conclusions in polynomial space and with an upper bound on complexity equal to P^{NP}. In contrast with DeLP and other argument-based approaches [4,8,21,22], the warrant computation algorithm for RP-DeLP is not based on the use of dialectical trees and it does not require to explicitly compute all possible arguments for a given literal in order to discover whether it is warranted, blocked or rejected.

2 Argumentation in RP-DeLP: An Overview

In order to make this paper self-contained, we will present next the main definitions that characterize the RP-DeLP framework. For further details the reader is referred to [2].

The language of RP-DeLP, denoted $\mathcal{L}_R$, is inherited from the language of logic programming, including the notions of atom, literal, rule and fact. Formulas are built over a finite set of propositional variables $p, q, ...$ which is extended with a new (negated) atom "$\sim p$" for each original atom p. Atoms of the form p or $\sim p$ will be referred as literals, and if P is a literal, we will use $\sim P$ to denote $\sim p$ if P is an atom p, and will denote p if P is a negated atom $\sim p$. Formulas of $\mathcal{L}_R$ consist of rules of the form $Q \leftarrow P_1 \wedge \ldots \wedge P_k$, where $Q, P_1, \ldots, P_k$ are literals. A fact will be a rule with no premises. We will also use the name *clause* to denote a rule or a fact. The RP-DeLP framework is based on the propositional logic $(\mathcal{L}_R, \vdash_R)$ where the inference operator $\vdash_R$ is defined by instances of the modus ponens rule of the form: $\{Q \leftarrow P_1 \wedge \ldots \wedge P_k, P_1, \ldots, P_k\} \vdash_R Q$. A set of clauses Γ will be deemed as *contradictory*, denoted $\Gamma \vdash_R \bot$, if , for some atom q, $\Gamma \vdash_R q$ and $\Gamma \vdash_R \sim q$.

A RP-DeLP program $\mathcal{P}$ is a tuple $\mathcal{P} = (\Pi, \Delta, \preceq)$ over the logic $(\mathcal{L}_R, \vdash_R)$, where $\Pi, \Delta \subseteq \mathcal{L}_R$, and $\Pi \nvdash_R \bot$. Π is a finite set of clauses representing strict knowledge (information we take for granted it holds true), Δ is another finite set of clauses representing the defeasible knowledge (formulas for which we have reasons to believe they are true). Finally, $\preceq$ is a *suitable* total pre-order on the set of defeasible formulas Δ. Suitable means that this pre-order is representable by a necessity measure N defined on the set of formulas of $\mathcal{L}_R$. Namely, we assume there exists N such that $\varphi \preceq \psi$ iff $N(\varphi) \leq N(\psi)$ for each $\varphi, \psi \in \Delta \cup \Pi$, where N is a mapping $N : \mathcal{L}_R \rightarrow [0, 1]$ such

that $N(\top) = 1, N(\bot) = 0$, $N(\varphi \wedge \psi) = \min(N(\varphi), N(\psi))$, and further $N(\varphi) = 1$ iff $\Pi \vdash_R \varphi$. For the sake of a simpler notation we will often refer to numerical weights for defeasible clauses and arguments rather than to the pre-ordering $\preceq$.

The notion of *argument* is the usual one. Given a RP-DeLP program $\mathcal{P}$, an argument for a literal Q of $\mathcal{L}_R$ is a pair $\mathcal{A} = \langle A, Q\rangle$, with $A \subseteq \Delta$ such that $\Pi \cup A \nvdash_R \bot$, and A is minimal (w.r.t. set inclusion) such that $\Pi \cup A \vdash_R Q$. If $A = \emptyset$, then we will call $\mathcal{A}$ a *s-argument* (s for strict), otherwise it will be a *d-argument* (d for defeasible). We define the *strength of an argument* $\langle A, Q\rangle$, written $s(\langle A, Q\rangle)$, as follows[2]:

$$s(\langle A, Q\rangle) = 1 \text{ if } A = \emptyset, \text{ and } s(\langle A, Q\rangle) = \min\{N(\psi) \mid \psi \in A\}, \text{ otherwise.}$$

The notion of *subargument* is referred to d-arguments and expresses an incremental proof relationship between arguments which is defined as follows. Let $\langle B, Q\rangle$ and $\langle A, P\rangle$ be two d-arguments such that the minimal sets (w.r.t. set inclusion) $\Pi_Q \subseteq \Pi$ and $\Pi_P \subseteq \Pi$ such that $\Pi_Q \cup B \vdash_R Q$ and $\Pi_P \cup A \vdash_R P$ verify that $\Pi_Q \subseteq \Pi_P$. Then, $\langle B, Q\rangle$ is a *subargument* of $\langle A, P\rangle$, written $\langle B, Q\rangle \sqsubset \langle A, P\rangle$, when either $B \subset A$ (strict inclusion for defeasible knowledge), or $B = A$ and $\Pi_Q \subset \Pi_A$ (strict inclusion for strict knowledge). A literal Q of $\mathcal{L}_R$ is called *justifiable* w.r.t. $\mathcal{P}$ if there exists an argument for Q, i.e. there exists $A \subseteq \Delta$ such that $\langle A, Q\rangle$ is an argument.

The following notion of acceptable argument with respect to a set (possibly empty) of justifiable conclusions W plays a key role to formalize the recursive warrant semantics. If we think of W of a consistent set of already warranted conclusions, an acceptable argument captures the idea of an argument which is based on subarguments already warranted. Let W be a set of justifiable conclusions which is consistent w.r.t. Π, i.e. $\Pi \cup W \nvdash_R \bot$. A d-argument $\mathcal{A} = \langle A, Q\rangle$ is an *acceptable argument* for Q w.r.t. W iff:

1. if $\langle B, P\rangle$ is a subargument of $\langle A, Q\rangle$ then $P \in W$
2. $\Pi \cup W \cup \{Q\} \nvdash_R \bot$

The usual notion of attack or defeat relation in an argumentation system is binary. However in certain situations, the conflict relation among arguments is hardly representable as a binary relation. For instance, consider the following RP-DeLP program $\mathcal{P} = (\Pi, \Delta, \preceq)$ with $\Pi = \{\sim p \leftarrow a \wedge b\}$ and $\Delta = \{a, b, p\}$, and consider just one level for defeasible information Δ. Clearly, $\mathcal{A}_1 = \langle\{p\}, p\rangle, \mathcal{A}_2 = \langle\{b\}, b\rangle, \mathcal{A}_3 = \langle\{a\}, a\rangle$ are arguments that justify p, b and a respectively, and which do not pair-wisely generate a conflict. Indeed, $\Pi \cup \{a, b\} \nvdash_R \bot$, $\Pi \cup \{a, p\} \nvdash_R \bot$ and $\Pi \cup \{b, p\} \nvdash_R \bot$. However the three arguments are collectively conflicting since $\Pi \cup \{a, b, p\} \vdash_R \bot$, hence in this program $\mathcal{P}$ there is a non-binary conflict relation among several arguments. Next we formalize this notion of *collective conflict among acceptable arguments* and which arises when we compare them with the strict part of a RP-DeLP program.

Let $\mathcal{P} = (\Pi, \Delta, \preceq)$ be a RP-DeLP program, let W be a consistent set of justifiable conclusions w.r.t. Π and let $\mathcal{A}_1 = \langle A_1, L_1\rangle, \ldots, \mathcal{A}_k = \langle A_k, L_k\rangle$ be acceptable arguments w.r.t. W of a same strength, i.e. such that $s(\langle A_1, L_1\rangle) = \ldots = s(\langle A_k, L_k\rangle)$. We say that the set of arguments $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ generates a conflict w.r.t. W iff the two following conditions hold:

[2] Actually, several necessity measures N may lead to a same pre-order $\preceq$, but we can take any of them to define the degree of strength since only the relative ordering is what matters.

(C) The set of argument conclusions $\{L_1, \ldots, L_k\}$ is contradictory w.r.t. $\Pi \cup W$, i.e. $\Pi \cup W \cup \{L_1, \ldots, L_k\} \vdash_R \bot$.
(M) The set $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ is minimal w.r.t. set inclusion satisfying (C), i.e. if $S \subset \{L_1, \ldots, L_k\}$, then $\Pi \cup W \cup S \nvdash_R \bot$.

In the above example, arguments $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ are acceptable w.r.t. Π and the empty set of conclusions $W = \emptyset$ and, according to our definition, it is clear that the set of acceptable arguments $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ for p, b and a respectively, generates a collective conflict (with respect to $W = \emptyset$). The intuition is that this collective conflict should block the conclusions a, b and p to be warranted, and therefore, for instance, $\mathcal{A}_4 = \langle \{a, b\}, \sim p \rangle$ is an argument for $\sim p$, but $\sim p$ should be a rejected conclusion since $\mathcal{A}_4$ is not acceptable w.r.t. $W = \emptyset$ because $\mathcal{A}_2$ and $\mathcal{A}_3$ are subarguments of $\mathcal{A}_4$ but obviously $a, b \notin W$. This general notion of collective conflict is used to define a recursive semantics for warranted conclusions of a RP-DeLP program.

Intuitively we define an output of a RP-DeLP program $\mathcal{P} = (\Pi, \Delta, \preceq)$ as a pair $(Warr, Block)$ of warranted and blocked conclusions (literals) respectively, all of them based on warranted information but, while warranted conclusions do not generate any conflict, blocked conclusions do. Because we are considering several levels of strength among arguments, the construction of the sets of conclusions *Warr* and *Block* is done level-wise, starting from the highest level and iteratively going down from one level to next level below. If $1 > \alpha_1 > \ldots > \alpha_p \geq 0$ are the strengths of d-arguments that can be built within $\mathcal{P}$, we define *d-Warr* $= \{$*d-Warr*$(\alpha_1), \ldots,$ *d-Warr*$(\alpha_p)\}$ and *Block* $= \{$*Block*$(\alpha_1), \ldots,$ *Block*$(\alpha_p)\}$, where *d-Warr*(α_i) and *Block*(α_i) are respectively the sets of the warranted and blocked justifiable conclusions with strength α_i. In the following, we write *d-Warrr*$(> \alpha_i)$ to denote $\cup_{\beta > \alpha_i}$*d-Warr*(β), and analogously for *Block*$(> \alpha_i)$, taking *d-Warr*$(> \alpha_1) = \emptyset$ and *Block*$(> \alpha_1) = \emptyset$.

Formally, an *output for a RP-DeLP program* $\mathcal{P} = (\Pi, \Delta, \preceq)$ is any pair (*Warr*, *Block*), where *Warr* = *s-Warr* $\cup$ *d-Warr* with *s-Warr* $= \{Q \mid \Pi \vdash_R Q\}$, such that *d-Warr* and *Block* are required to satisfy the following recursive constraints:

1. A d-argument $\langle A, Q \rangle$ of strength α_i is called *valid* (or not rejected) if it satisfies the following three conditions[3]:
   (i) for every subargument $\langle B, P \rangle \sqsubset \langle A, Q \rangle$ of strength α_i, $P \in$ *d-Warr*(α_i);
   (ii) $\langle A, Q \rangle$ is acceptable w.r.t. $W =$ *d-Warr*$(> \alpha_i) \cup \{P \mid \langle B, P \rangle \sqsubset \langle A, Q \rangle$ *and* $s(\langle B, P \rangle) = \alpha_i\}$;
   (iii) $Q \notin$ *d-Warr*$(> \alpha_i) \cup$ *Block*$(> \alpha_i)$ and $\sim Q \notin$ *Block*$(> \alpha_i)$;
2. For every valid argument $\langle A, Q \rangle$ of strength α_i we have that
   – $Q \in$ *d-Warr*(α_i) whenever there does not exist a set G of valid arguments of strength α_i such that
     (i) $\langle A, Q \rangle \not\sqsubset \langle C, R \rangle$ for all $\langle C, R \rangle \in G$
     (ii) $G \cup \{\langle A, Q \rangle\}$ generates a conflict w.r.t. $W =$ *d-Warr*$(> \alpha_i) \cup \{P \mid$ there exists $\langle B, P \rangle \sqsubset \langle D, L \rangle$ for some $\langle D, L \rangle \in G \cup \{\langle A, Q \rangle\}\}$
   – otherwise, $\varphi \in$ *Block*(α_i).

[3] Notice that if $\langle A, Q \rangle$ is a d-argument, $A \neq \emptyset$ and, because of the notion of argument, $\Pi \nvdash_R Q$ and hence $Q \notin$ *s-Warr*. Moreover, if $\langle A, Q \rangle$ is an acceptable d-argument w.r.t. *d-Warr*$(> \alpha_i)$, then $\langle A, Q \rangle$ is valid whenever condition (iii) holds.

The intuition underlying this definition is as follows: a d-argument $\langle A, Q\rangle$ of strength α_i is either warranted or blocked whenever for every subargument $\langle B, P\rangle$ of $\langle A, Q\rangle$, P is warranted and there does not exist a different valid argument for Q of strength greater than α_i; then, it is eventually warranted if Q is not involved in any conflict, otherwise it is blocked.

For instance, consider the RP-DeLP program $\mathcal{P}$ of the previous example with $\Pi = \{\sim p \leftarrow a \wedge b\}$ and $\Delta = \{a, b, p\}$ extended with three levels of defeasibility as follows: $p \prec b \prec a$. Assume α_1, α_2 and α_3 are the levels of a, b and c respectively, obviously with $1 > \alpha_1 > \alpha_2 > \alpha_3$. Then, *s-Warr* $= \emptyset$ and the argument for $\langle\{a\}, a\rangle$ is the only valid argument with strength α_1. Then, at level α_1, we get *d-Warr*$(\alpha_1) = \{a\}$ and *Block*$(\alpha_1) = \emptyset$. At level α_2, we have that $\langle\{b\}, b\rangle$ is a valid argument w.r.t. *d-Warr*(α_1) because $\Pi \cup \{a\} \cup \{b\} \not\vdash_R \bot$, and $\langle\{b\}, b\rangle$ does not produce any conflict at level α_2. Then, b is a warranted conclusion at level α_2, and thus, $\langle\{a, b\}, \sim p\rangle$ is also a valid argument w.r.t. *d-Warr*$(\alpha_1) \cup \{b\}$. Hence, at level α_2 we get *d-Warr*$(\alpha_2) = \{b, \sim p\}$ and *Block*$(\alpha_1) = \emptyset$. Finally, at level α_3 the argument $\langle\{p\}, p\rangle$ for p is not valid w.r.t. *d-Warr*$(\alpha_1) \cup$ *d-Warr*(α_2) since $\Pi \cup \{a\} \cup \{b, \sim p\} \cup \{p\} \vdash_R \bot$, and thus, at level α_3 we get *d-Warr*(α_3) = *Block*$(\alpha_3) = \emptyset$.

It can be proven that if (*Warr*, *Block*) is an output for a RP-DeLP program $\mathcal{P} = (\Pi, \Delta, \preceq)$, the set *Warr* of warranted conclusions is indeed non-contradictory, satisfies the indirect consistency property ($\Pi \cup$ *Warr* $\not\vdash_R \bot$), and it is closed with respect to the strict knowledge (if $\Pi \cup$ *Warr* $\vdash_R Q$ then $Q \in$ *Warr*).

In [2] we showed that, in some cases, a RP-DeLP program may have multiple outputs $(Warr, Block)$ due to some circular definitions of warranty that emerge when considering conflicts among arguments. Such circular definitions of warranty are identified by means of what we called *warrant dependency graph* of a RP-DeLP program. Intuitively, the warrant dependency graph for a set of arguments represents conflict and support dependences among arguments with respect to a set of justified conclusions.

For instance, consider a RP-DeLP program with an empty set of strict clauses and the set of defeasible clauses $\Delta = \{p, q, \sim p \leftarrow q, \sim q \leftarrow p\}$ with just one defeasibility level. Obviously, *s-Warr* $= \emptyset$ and the arguments $\mathcal{A}_1 = \langle\{p\}, p\rangle$ and $\mathcal{A}_2 = \langle\{q\}, q\rangle$ for conclusions p and q, respectively, are valid arguments with respect to *s-Warr*. Now consider the arguments for conclusions $\sim p$ and $\sim q$; i.e. $\mathcal{B}_1 = \langle\{q,\ \sim p \leftarrow q\}, \sim p\rangle$ and $\mathcal{B}_2 = \langle\{p,\ \sim q \leftarrow p\}, \sim q\rangle$. In this example, the arguments $\mathcal{A}_1$ and $\mathcal{A}_2$ are valid arguments, and thus, conclusions p and q may be warranted or blocked but not rejected. Moreover, the argument $\mathcal{B}_1$ may be valid whenever the conclusion q is warranted, and the argument $\mathcal{B}_2$ may be valid whenever the conclusion q is warranted. However, if the argument $\mathcal{B}_1$ is valid, then p and $\sim p$ are blocked conclusions, and if the argument $\mathcal{B}_2$ is valid, then q and $\sim q$ are blocked conclusions. Hence, in that case we have two possible outputs: $(Warr_1, Block_1)$ with $Warr_1 = \{p\}$ and $Block_1 = \{q, \sim q\}$, and $(Warr_2, Block_2)$ with $Warr_2 = \{q\}$ and $Block_2 = \{p, \sim p\}$.

3 On the Computation of the Unique Output

From a computational point of view, whether a RP-DeLP programs has a unique output can be checked by means of a level-wise procedure, starting from the highest level and

iteratively going down from one level to next level below, and for every level verifying that there is no cycle in the graph for every valid argument with respect to the set of warranted conclusions at previous levels and the current level. Next we define an algorithm which implements this level-wise procedure computing warranted and blocked conclusions until a cycle is found or the unique output is obtained. In the following we use the notation $W(1)$ for *s-Warr*, $W(\alpha)$ for *d-Warr*(α) and $B(\alpha)$ for *Block*(α). Then, W denotes $\cup_{1\geq\alpha>0}W(\alpha)$, B denotes $\cup_{1>\alpha>0}B(\alpha)$, $W(\geq\alpha)$ denotes $\cup_{\beta\geq\alpha}W(\beta)$ and $B(\geq\alpha)$ denotes $\cup_{\beta\geq\alpha}B(\beta)$.

Algorithm `Computing warranted conclusions`
Input $\mathcal{P} = (\Pi, \Delta, \preceq)$: a RP-DeLP program
Output
$unicity$: Boolean value for the unique output property
(W, B): unique output for $\mathcal{P}$
Method
$unicity$:= True
$W(1) := \{Q \mid \Pi \vdash_R Q\}$
$B := \emptyset$
$\alpha :=$ `maximum_defeasibility_level`(Δ)
while ($unicity$ and $\alpha > 0$)
    `level_computing`$(\alpha, W, B, unicity)$
    $\alpha :=$ `next_defeasibility_level`(Δ)
end while
end algorithm `Computing warranted conclusions`

The algorithm `Computing warranted conclusions` first computes the set of warranted conclusions $W(1)$ form the set of strict clauses Π. Then, for each defeasibility level $1 > \alpha > 0$, the procedure `level_computing` determines all warranted and blocked conclusions with necessity degree α whenever there does not exist a circular warranty definition between arguments.

Procedure `level_computing` (**in** α; **in_out** W, B, $unicity$)
VC : = $\{Q \mid \langle C, Q\rangle$ with strength α is valid w.r.t. $(W, B)\}$
while ($unicity$ and *VC* $\neq \emptyset$)
    while ($\exists\ Q \in$ *VC* $\mid \neg$ `conflict`$(\alpha, Q,$ *VC*$, W,$ `not_depend`$(\alpha, Q,$ *VC*$, W, B))$)
        $W(\alpha) := W(\alpha) \cup \{Q\}$
        VC := *VC*$\backslash\{Q\} \cup \{P \mid \langle C, P\rangle$ with strength α is valid w.r.t. $(W, B)\}$
    end while
    $I := \{Q \in$ *VC* $\mid$ `conflict`$(\alpha, Q,$ *VC*$, W, \emptyset)\}$
    $B(\alpha) := B(\alpha) \cup I$
    VC := *VC*$\backslash I$
    if (for some $Q \in$ *VC* there is a cycle in the graph w.r.t. W) **then** $unicity$:= False
end while
end procedure `level_computing`

For every level α the procedure `level_computing` computes the set of valid conclusions *VC* with respect to the current sets of warranted and blocked conclusions (W, B)[4]. The set of valid conclusions *VC* is dynamically updated depending on new warranted conclusions. The procedure `level_computing` is based on the two following auxiliary functions.

Function `conflict`(**in** α, Q, *VC*, W, D): **return Boolean**
return ($\exists$ $S \subseteq VC \backslash \{Q\} \cup D$ such that $\Pi \cup W(\geq \alpha) \cup S \nvdash_R \bot$
and $\Pi \cup W(\geq \alpha) \cup S \cup \{Q\} \vdash_R \bot$)
end function `conflict`

Function `not_depend`(**in** α, Q, *VC*, W, B): **return** set of conclusions
D := $\{P \notin VC \mid \langle C, P\rangle$ with strength α is almost valid w.r.t. (W, B)
and literal Q is not a conclusion in $C\}$
return(D)
end function `not_depend`

The function `conflict` checks conflicts among Q, the set of valid conclusions $VC\backslash\{Q\}$ and the set of conclusions D. The set of conclusions D takes two different values: the empty set and what we call *almost valid* conclusions with respect to (W, B)[5] which do not depend on Q; i.e. conclusions which depend on some valid conclusion in $VC\backslash\{Q\}$, do not depend on blocked conclusions and do not generate conflicts. In fact, the function `not_depend` computes the set of conclusions D for a given literal $Q \in VC$. Finally, we remark that the existence of a cycle in the graph for some $Q \in VC$ with respect to the current set of warranted conclusions W can be determined by checking the stability of *VC* after two consecutive iterations instead of explicitly building the warranty dependence graph for every $Q \in VC$, since the function `conflict` considers recursive warranty definitions among arguments by means of the set of conclusions D.

One of the main advantages of the warrant recursive semantics for RP-DeLP is from the implementation point of view. Actually, warrant semantics based on dialectical trees and, in general, rule-based argumentation frameworks like DeLP [7,9], might consider an exponential number of arguments with respect to the number of rules of a given program. In contrast, in our framework, at least for the particular case of RP-DeLP programs with unique output, it is not necessary to explicitly compute all the possible arguments for a given literal, in order to discover whether it is warranted, as we can implement the previous algorithm with a worst-case complexity in P^{NP}.

First, observe that the incremental discovery of valid conclusions (set *VC* in the algorithm) can be performed in P-time, as for every defeasibility level α all that we need

[4] Notice that for every level of execution α, an argument $\langle C, Q\rangle$ with strength α is *valid* w.r.t. the current sets (W, B) iff (i) $Q \notin W(\geq \alpha)$ and $Q, \sim Q \notin B(\geq \alpha)$, and (ii) $\langle C, Q\rangle$ is acceptable w.r.t. $W(\geq \alpha)$.

[5] For every level of execution α, an argument $\langle C, P\rangle$ with strength α is called *almost valid* w.r.t. the current sets (W, B) iff (i) $P \notin W(\geq \alpha)$ and $P, \sim P \notin B(\geq \alpha)$; (ii) for all $\langle E, R\rangle \sqsubset \langle C, P\rangle$ with strength $\beta > \alpha$, $R \in W(\beta)$; (iii) for all $\langle E, R\rangle \sqsubset \langle C, P\rangle$ with strength α, $R, \sim R \notin B(\geq \alpha)$; (iv) for some $\langle E, R\rangle \sqsubset \langle C, P\rangle$ with strength α, $R \notin W(\alpha)$; and (v) $\Pi \cup W(\geq \alpha) \cup \{R \mid \langle E, R\rangle \sqsubset \langle C, P\rangle$ with strength $\alpha\} \cup \{P\} \nvdash_R \bot$.

for a literal Q to be in *VC* is either that $Q \in \Delta$ with level α, or an α-rule with warranted body and conclusion Q. An $\alpha-$rule R is a rule with a level greater or equal to α but with $Body(R) \setminus W(\geq \alpha) \neq \emptyset$, so that it can only conclude its conclusion with strength equal or less than α. Secondly, remember that we can effectively avoid building and checking the dependency graph for literals, as if the program has unique output, the set *VC* changes at every iteration of the main loop at `level_computing`. Finally, we need only to check the complexity of the following problems:

1. Whether a literal P is in the set D = `not_depend`(α, Q, *VC*, W, B). We can non-deterministically guess a subset $S \subseteq (W(\geq \alpha) \cup \Delta(\alpha))$ and check in polynomial time whether it encodes an almost valid argument for P without using Q. The set $\Delta(\alpha)$ is the set of facts with level α and $\alpha-$rules, as defined before. So, we can check whether P has such almost valid argument with an NP algorithm[6].
2. Whether the function `conflict`(α, Q, *VC*, W, D) returns true. Remark that we can non-deterministically guess a subset $S \subseteq VC \setminus \{Q\} \cup D$ and check in polynomial time whether $\Pi \cup W(\geq \alpha) \cup S \not\vdash_R \bot$ and $\Pi \cup W(\geq \alpha) \cup S \cup \{Q\} \vdash_R \bot$). So, again this can be checked with an NP algorithm.

Finally, these basic problems are solved at most a polynomial number of times in the `level_computing` procedure. In `level_computing` a literal is never inserted again in the set *VC* once it has been removed from it, and the number of iterations of the inner loop, the one that discovers warranted literals, is bounded by the size of *VC*. So, given that the outer loop of `level_computing` will end as soon as no more literals are found in *VC* or as soon as *VC* becomes stable (because no valid conclusions can be either warranted or blocked), the number of steps of the procedure is polynomially bounded by the size of the program. Also, the number of times that `level_computing` is called is bounded by the number of levels of defeasibility. So, this gives an upper bound on complexity equal to P^{NP} for discovering whether a literal is warranted in the unique output, or a function complexity of FP^{NP} for computing the unique output.

It is worth noticing that in a recent work [18], the authors have studied the complexity of the warranted formula problem in a framework for propositional argumentation with classical logic and general formulae (not only horn clauses), and they have shown the problem to be PSPACE-complete. In our case, at least for the particular case of RP-DeLP programs with unique output, the complexity is upper bounded with a subclass of PSPACE, P^{NP}, but we suspect that for general RP-DeLP programs the complexity of determining whether a literal is in the skeptical output is at least PSPACE-hard. In contrast, in abstract argumentation frameworks similar reasoning problems have lower complexity than for propositional argumentation. For example, checking whether an argument is included on some preferred extension is NP-complete [10] and checking whether an argument is included on every preferred extension (skeptical reasoning) is $coNP^{NP}$-complete [15]. Recently, the related problem of checking whether an

[6] The argument found by such NP algorithm may not be minimal, but as we are concerned with the existence problem, if there is an almost valid argument, then a subset of it will be a minimal one. It can be shown that the main algorithm only stores a literal in the warrant set once it has found a minimal argument that is conflict free.

argument is in the maximal ideal extension has been shown to be between coNP and a subclass of P^{NP} [16,14].

4 Conclusions and Future Work

In this paper we have designed an algorithm for computing the warranty status of arguments according to the new recursive semantics defined in [2] for defeasible argumentation with defeasibility levels, for the particular case of programs with unique output. It discovers warranted literals using a polynomial amount of memory, as it avoids maintaining sets of arguments that could be of exponential size with respect to program size, and its worst-case complexity is upper bounded with the class P^{NP}. Moreover, taking advantage of the recursive semantics, it allows to recover the argument for every warranted literal and even for blocked literals it is possible to store the conflict sets that were found as responsible of their blocking. So in possible applications of this framework this algorithm could provide useful information for users in case of unexpected outputs.

Future work will be addressed in two main directions. On the one hand we plan to design an efficient version of the algorithm we have shown here, by minimizing the effective number of NP queries that have to be made during the execution of the `level_computing` procedure. There are several ways of saving NP queries. For example, every time an almost valid argument has been found for a literal Q, this argument can be saved, so next time we need to check an argument for Q, if the one saved is still almost valid (i.e. it is not based on blocked conclusions), there is no need to find a new one. Also, with the aim of obtaining an algorithm able to scale up with problem size, we will design polynomial time reductions of the NP queries to be performed to the SAT problem, so that we can take profit of state-of-the-art SAT solvers for solving the most critical subproblems during the search. We also plan to study particular cases of RP-DeLP programs that have a better worst-case complexity, looking for tractable cases, like it has been done recently for other argumentation frameworks, like the case of bipartite abstract argumentation frameworks [14].

On the other hand we plan to come to the problem of deciding which output should be considered for RP-DeLP programs with multiple outputs. A natural solution to this problem could be to adopt the intersection of all possible outputs in order to define the skeptical output as the set of those literals which are ultimately warranted. However, as stated in [19], adopting the intersection of all possible outputs can lead to an inconsistent output when some recursive situation occurs between the literals of a program. So that we plan to define an ultimately warranted conclusion for RP-DeLP programs with multiple outputs as a conclusion of the intersection which is recursively based on ultimately warranted conclusions. In fact, this idea corresponds with the maximal ideal extension defined by Dung, Mancarella and Toni [12,13] as an alternative skeptical basis for defining collections of justified arguments in the abstract argumentation frameworks promoted by Dung [11] and Bondarenko *et al.* [5].

Acknowledgments. Authors are thankful to the anonymous reviewers for their helpful comments. Research partially funded by the Spanish MICINN projects MULOG2 (TIN2007-68005-C04-01/02) and ARINF (TIN2009-14704-C03-01/03),

CONSOLIDER (CSD2007-0022), and ESF Eurocores-LogICCC/MICINN (FFI2008-03126-E/FILO), and the grant JC2009-00272 from the Ministerio de Educación.

References

1. Alsinet, T., Chesñevar, C.I., Godo, L., Simari, G.: A logic programming framework for possibilistic argumentation: Formalization and logical properties. Fuzzy Sets and Systems 159(10), 1208–1228 (2008)
2. Alsinet, T., Béjar, R., Godo, L.: A characterization of collective conflict for defeasible argumentation. In: Proc. of COMMA 2010, pp. 27–38 (2010)
3. Alsinet, T., Chesñevar, C.I., Godo, L.: A Level-based Approach to Computing Warranted Arguments in Possibilistic Defeasible Logic. In: Proc. of COMMA 2008, pp. 1–12 (2008)
4. Besnard, P., Hunter, A.: Elements of Argumentation. The MIT Press, Cambridge (2008)
5. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artif. Intell. 93, 63–101 (1997)
6. Caminada, M., Amgoud, L.: On the evaluation of argumentation formalisms. Artif. Intell. 171(5-6), 286–310 (2007)
7. Cecchi, L., Fillottrani, P., Simari, G.: On the complexity of DeLP through game semantics. In: Proc. of NMR 2006, pp. 386–394 (2006)
8. Chesñevar, C.I., Maguitman, A., Loui, R.: Logical Models of Argument. ACM Computing Surveys 32(4), 337–383 (2000)
9. Chesñevar, C.I., Simari, G., Godo, L.: Computing dialectical trees efficiently in possibilistic defeasible logic programming. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 158–171. Springer, Heidelberg (2005)
10. Dimopoulos, Y., Torres, A.: Graph theoretical structures in logic programs and default theories. Theoretical Computer Science 170(1-2), 209–244 (1996)
11. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–358 (1995)
12. Dung, P.M., Mancarella, P., Toni, F.: A dialectic procedure for sceptical, assumption-based argumentation. In: Proc. of COMMA 2006, pp. 145–156 (2006)
13. Dung, P.M., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artif. Intell. 171(10-15), 642–674 (2007)
14. Dunne, P.E.: The computational complexity of ideal semantics. Artif. Intell. 173(18), 1559–1591 (2009)
15. Dunne, P.E., Bench-Capon, T.J.M.: Coherence in finite argument systems. Artif. Intell. 141(1-2), 187–203 (2002)
16. Dunne, P.E.: The computational complexity of ideal semantics i: Abstract argumentation frameworks. In: Proc. of COMMA 2008, pp. 147–158 (2008)
17. García, A., Simari, G.: Defeasible Logic Programming: An Argumentative Approach. Theory and Practice of Logic Programming 4(1), 95–138 (2004)
18. Hirsch, R., Gorogiannis, N.: The complexity of the warranted formula problem in propositional argumentation. J. of Logic and Computation 20(2) (2009)
19. Pollock, J.L.: A recursive semantics for defeasible reasoning. In: Rahwan, I., Simari, G. (eds.) Argumentation in Artificial Intelligence. ch. 9, pp. 173–198. Springer, Heidelberg (2009)
20. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. J. of Applied Non-classical Logics 7, 25–75 (1997)
21. Prakken, H., Vreeswijk, G.: Logical Systems for Defeasible Argumentation. In: Gabbay, D., Guenther, F. (eds.) Handbook of Phil. Logic, pp. 219–318. Kluwer, Dordrecht (2002)
22. Rahwan, I., Simari, G. (eds.): Argumentation in Artificial Intelligence. Springer, Heidelberg (2009)

Improving Efficiency of a Genetic Algorithm Applied to Multi-robot Tactic Operation

Gustavo Pessin, Fernando S. Osório, Denis F. Wolf, and Christiane R.S. Brasil

University of São Paulo (USP)
Institute of Mathematics and Computer Science (ICMC)
Av. Trabalhador Sãocarlense, 400 – C.P. 668 – 13.560-970 – São Carlos, SP
{pessin,fosorio,denis,christiane}@icmc.usp.br

Abstract. There are two important issues in the Genetic Algorithm searching and optimization process: population diversity and selective pressure. These two issues are strongly related and have direct impact on the search efficiency. Two factors that directly influence these issues are often ignored: overlapping populations and fitness scaling. In this paper we address the use of overlapping populations and fitness scaling in a Genetic Algorithm (GA) applied to multi-robot squad formation and coordination. The robotic task is performed over a natural disaster scenario (a forest fire simulation). The robot squad mission is surrounding the fire and avoiding fire's propagation based on the strategy proposed by the GA. Simulations have been carried out with several GA parameters (several types of scaling and different degrees of overlapping) in order to obtain the most efficient optimization for group formation and task execution. Simulations results show that the use of overlapping population and fitness scaling present better results than non-overlapping population and unscaled fitness.

Keywords: Genetic algorithm, multi-robot systems, coordination.

1 Introduction

There are many fields where a single agent is not sufficient or enough to fulfill a determined task. Tasks like cleaning nuclear residuals, cleaning chemical accidents, forest fire combat or even on constructions, agriculture, hostile environment exploration, security and critical missions may be better accomplished when using a group of agents. Using robotic agents instead of human beings may add security, reliability and efficiency in these tasks. Multirobotic systems are extremely dependent on control techniques; they can add scalability, flexibility and robustness to a wide range of new applications [1,17,19], but they also bring a series of new questions to be solved in collaboration and cooperation. Specialized algorithms, composed by rules and automats have been developed seeking to coordinate these physical sets in dynamic environments, showing to be an extremely complex challenge [7,3]. Due to this, a large number of researchers are

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 50–59, 2010.

migrating their efforts to several different approaches (*e.g.* application of classical intelligent artificial techniques, social models, market-based models, swarm-based models).

In the firefighting mission, one of the most important questions is related to the robot position setting. According to the actuation capability of each robot, weather condition (wind, rain), topography, and vegetation, several arrangements can be proposed. These arrangements, when suggested by a specialist, may not take in account a large number of variables. In these cases, machine learning techniques may be successfully used. One of the machine learning techniques that has been showing satisfactory results in solving optimization problems are Genetic Algorithms [11,16]. It consists in a global optimization algorithm that employs parallel and structured search strategies, directed by fitness points seeking, allowing the multi-criteria search in a multidimensional space. As it is an unsupervised technique, GAs don't need any previous information or solution database. Monitoring and combating of forest fire is an example of multirobotic system that could considerably reduces human, material and environmental losses.

In [21] we proposed and evaluated a Genetic Algorithm to accomplish the formation of a robotic squad that should perform a firefighting task; we evaluate characteristics like chromosome structure, mutation rate, types of mutation, number of individuals and crossover rate. However, two factors that has a strongly influence in the efficiency of the genetic search were not examined, which are: overlapping populations and scaling fitness. Thus, in this paper, we extended the work [21] to evaluate the use of overlapping populations and scaling fitness. The goal of this paper is to find a new set of parameters that allows that the GA converges more efficiently.

This paper has the following structure: Section 2 introduces short theoretical description of robot's applications. Section 3 presents concepts and applications of GAs. In Section 4 we explain the developed environment, the proposed fitness and the chromosome's structure. Section 5 describes the evaluation of all performed experiments. We finalize presenting the conclusion of the presented work and the future perspectives.

2 Mobile Robotics

Several current works demonstrate mobile robotic usage as individual systems on hostile operations as the rescue auxiliary robot Raposa [13] and SACI robot [14] developed for acting on fire combat. Moreover, there are robots to perform tasks on aquatic environments, space, caves and volcanoes exploration, and even to household use. Multirobotic systems must be formed by robots that are able to effective act on tasks, so knowledge about robotic control is a very important field. Works describing intelligent robot navigation can be seen in [25,10]. In 2004 and 2005, DARPA Grand Challenge [4], financed by the Defense Advanced Research Projects Agency organized a competition where the goal was building a completely autonomous vehicle that could complete a long way on dirt road on limited time. In 2007 the focus of the competition has changed. Renamed to

DARPA Urban Challenge, it had a new goal to develop a vehicle that could perform autonomously on urban traffic, and execute tasks like parking, overtaking and intersection negotiations. These examples show trends in cooperation and multiple interactions.

The work with groups adds a great number of possibilities on tasking-solving but brings a series of new questions to be solved in collaboration and cooperation. Works using multirobotic systems like [24,2] uses pre-programmed rules on agents to perform formation. In [17,6] techniques are explored to perform works with collectives robotics, used mainly for the purpose of applying the concept of self-organization and collective optimization although task division is not directed explored. The works described in this section demonstrate that the application of mobile robotics in control of incidents is an important and active topic of research and development. These several competitions also demonstrate that there is still not a definitive or more adequate solution to the problem, and it is an open research field. In all consulted documents there is no consensual form to multirobotic system's conformation and actuation. Unpredicted situations with large degree of autonomy and robustness are still difficult to handle.

3 Genetic Algorithms

Genetic Algorithms (GA) [11,16] are global optimization techniques that employ parallel and structured search strategies, allowing multi-criteria search in a multi-dimensional space. They are methods classified as unsupervised, being unnecessary any previous information database. The GAs use iterative procedures that simulate the evolution process of a population constituted by candidate solutions of a certain problem. The evolution process is guided by a selection mechanism based on fitness of individual structures. For each algorithm iteration (single generation), a new structure set is created by information changing (bit or blocks) between selected structures of the previous generation [8]. The result of this conduces to the increasing individual fitness. A GA is structured in a way that the information about a determined system can be coded similarly to a biological chromosome, like a value array, where usually each sequence fragment represents a variable.

Dealing with applications, [18] uses a GA to satisfactorily optimize trajectory planning for a robot arm. In [9], a GA model correctly evolves values for force and time application to allow a robot to walk. The work [26] presents a GA model to evolve the exploration method of a mobile robot in an unknown environment. None of the works describes the use of overlapping population or fitness scaling. The works [12,23] show the use of overlapping population and fitness scaling in some theoretical problems (mathematical functions). Their results emphasize that the use of this issues are problem-dependent.

3.1 Genetic Issues

There are two important issues in the GA searching and optimization process: population diversity and selective pressure. These two issues are strongly related

and have direct impact on the search efficiency. An increase in the selective pressure decrease the population diversity and vice-versa [15]. The population diversity and selective pressure has direct impact on premature convergence or ineffective search. Thus, it is important to stipulate a balance between these factors. Sampling mechanisms are attempt to achieve this goal [15,23]. Two factors that have directly influence in the sampling mechanisms are overlapping populations and fitness scaling.

The technique of overlapping populations consists of creating a copy of the current population. The genetic operations (selection, crossover and mutation) are carried out on this copy. After the evaluation of individuals in this cloned population, the individuals of original population are replaced considering a percentage of overlap defined by the user. The worst individuals of the original population are replaced by the best cloned population. Thus, they are kept in the original population of a certain amount of unique individuals. The n best individuals have at least one more generation to accomplish for crossover and mutation. Replacing just one part of the population decreases the diversity of the population, thus the use of overlapping population converges faster. However, the exploitation of the search space is much more restricted [5,8].

Fitness scaling is done by applying a mathematical function on the original fitness, such that a sample of the population considers the scaled value instead of the raw fitness [23]. There are fitness functions where the value can not be really significant for the search with the GA. Thus, methods have been proposed in literature of scale for the fitness in the search that may increase or not selective pressure and consequently decrease or increase in the diversity of the population [8,15,23]. In this study, beyond the evaluation of the raw fitness, four types of scaling are evaluated:

- *Linear Scaling*: Normalizes the fitness based on minimum and maximum fitness in current population. Sometimes can have a scale factor. The scale factor of 1.2 has been used in accordance to [8];
- *Power Law Scaling*: Maps objective scores to fitness scores using an exponential relationship defined as $f_i^{'} = f_i^k$, such that $f_i^{'}$ is the original fitness and $f_i^{'}$ is the scaled fitness. Several studies have shown success using k = 1.005, so we use these values in our search;
- *Sigma Truncation Scaling*: Normalizes using population mean and standard deviation, as $f_i^{'} = f_i + (\overline{f} - c \cdot \sigma)$. We use $c = 2.0$ in accordance with specified in [22];
- *Sharing Scaling*: Reduces fitness for individuals that are similar to other individuals in the population. This scaling method is used to do speciation[1]. The fitness score is derived from its objective score by comparing the individual against the other individuals in the population. If there are other similar individuals then the fitness is derated [22].

[1] Speciation is the evolutionary process by which new biological species arise.

4 Group Formation

In order to build a real physical implementation of robotic system, it is highly recommended to test the algorithms on virtual realistic simulation environments. Robotic system's simulation is specially necessary in case of large, expensive or fragile robots because it is an powerful tool to avoid wasting resources [7]. In our case, the proposed simulator should be able to reproduce an environmental disaster for a multirobotic system actuation. We propose the situation of a forest fire. In this case a intelligent vehicle squad (as road grader) has the purpose of combating the forest fire acting by creating firebreaks around the fire.

The detailed characteristics of the fire spreading modeled to this work, as well as the forest fuel models and the real operation techniques are compiled into [20]. The simulated terrain is based on topographical maps and on forest fuel maps models that can also be seen in [20]. The fire spreading simulation try to model the fire propagation as realistic as possible (considering present vegetation type, terrain slope, wind orientation and intensity).

Table 1. Chromosome structure (group of four robots)

Gene	Function	Min. value	Max. value
0	Initial angle of robot 0	0.0°	360.0°
1	Final angle of robot 0; initial of robot 1	0.0°	360.0°
2	Final angle of robot 1; initial of robot 2	0.0°	360.0°
3	Final angle of robot 2; initial of robot 3	0.0°	360.0°
4	Final angle of robot 3	0.0°	360.0°
5	Initial radius of robot 0	10.0m	100.0m
6	Final radius of robot 0; initial of robot 1	10.0m	100.0m
7	Final radius of robot 1; initial of robot 2	10.0m	100.0m
8	Final radius of robot 2; initial of robot 3	10.0m	100.0m
9	Final radius of robot 3	10.0m	100.0m

The planning mechanism uses a GA to define the initial and final operation positions of each robot for fire-fighting, which is developed using GAlib library [22]. Considering that the combat agents are graders which have the finality to create firebreaks, we require that the GA returns the following information: initial and final angle, and initial and final radius for each robot, both related to the fire starting point. The proposed chromosome can be seen in the Table 1. In this, it is presented information of all group of involved agents, thus, the chromosome size depends on the number of robots in the system. Regarding the GAs parameters, we use alleles that limit the value set generated for each attribute (radius between 10.0 and 100.0 and angles between 0.0 and 360.0 degrees). The use of alleles reduces the search space. Also, we used real genome, optimized for floating point numbers.

The coordinates of operation are calculated using $x_d = x_a + r_i \cdot \cos(a_i)$ and $y_d = y_a + r_i \cdot \sin(a_i)$. Where (x_d, y_d) is the robot's destination position, (x_a, y_a) is the starting position of the fire, r_i is the radius (gene 5 to 9) and a_i is the angle (gene 0 to 4). The radius and the angle are defined specifically to each operation of each robot (initial and final coordinate of firebreaks creation).

The proposed fitness is related with saved vegetation area and combat units usage rate; therefore, the fitness accumulates: (i) Total burned area: trying to minimize burned area, (ii) Firebreak total area: trying to minimize robot's work area, avoiding to create firebreak on non-risk areas, (iii) Trying to minimize the difference among general average of useful firebreaks in relation to each individual useful firebreak, equalizing worked areas. The GA tries to minimize the fitness function value, that means less burned vegetation, less created firebreaks, and less difference between the size of firebreaks of each robot.

5 Experiments and Results

We evaluated the system considering different percentages of overlap as well as different scaling methods, as show Table 2. Another GA parameters were obtained from [21]: {selection by stochastic remainder sampling selector; uniform mutation; 10% of mutation; 90% of crossover; two point crossover; 150 individuals; 700 generations}. The climatic characteristics of the fire simulation and the initial positions of robots are set as fixed in order to make the simulations. Ten simulations were performed with each set of parameters, totaling 200 simulations. The simulations considered the existence of four combat agents.

Table 2. Parameter set evaluated on the GA

Set	Scaling Method	Overlapping
$A_1 - A_4$	No Scaling	{No; 25%; 50%; 75%}
$B_1 - B_4$	Linear	{No; 25%; 50%; 75%}
$C_1 - C_4$	Sigma	{No; 25%; 50%; 75%}
$D_1 - D_4$	Power	{No; 25%; 50%; 75%}
$E_1 - E_4$	Sharing	{No; 25%; 50%; 75%}

Simulations results can bee seen in Fig. 1. Many visual observations were done on the fire simulator (using the results of the GA), showing that fitness that are below 3,500 are considered satisfactory – the simulations with fitness below 3,500 are able to extinguish the fire with well distributed areas among the robots. Fig. 2 shows some satisfactory and unsatisfactory results of the simulations, where the Fig. 2(a) and 2(b) presents, respectively, fitness of 4,800 and 6,100; considered unsatisfactory results. The Fig. 2(c) present fitness of 3,200 units; considered satisfactory. We can see in Fig. 1 that only four sets present mean and standard deviation, for all simulations, below 3,500 units. These sets were: {linear scaling with 50% of overlapping; sharing scaling with 50% of overlapping; no scaling with 75% of overlapping and sharing scaling with 75% of overlapping}. Also, we can see that simulations that do not use overlapping populations present considerably worse values.

Whereas we obtained four sets with satisfactory results (fitness below 3,500), the next step was to seek improvements by reducing the number of individuals. Thus we performed a new set of simulations considering the four best sets of parameters but with a decrease in the number of individuals from 150 to 100 and 50. The size of the population is one of the most important choices faced by

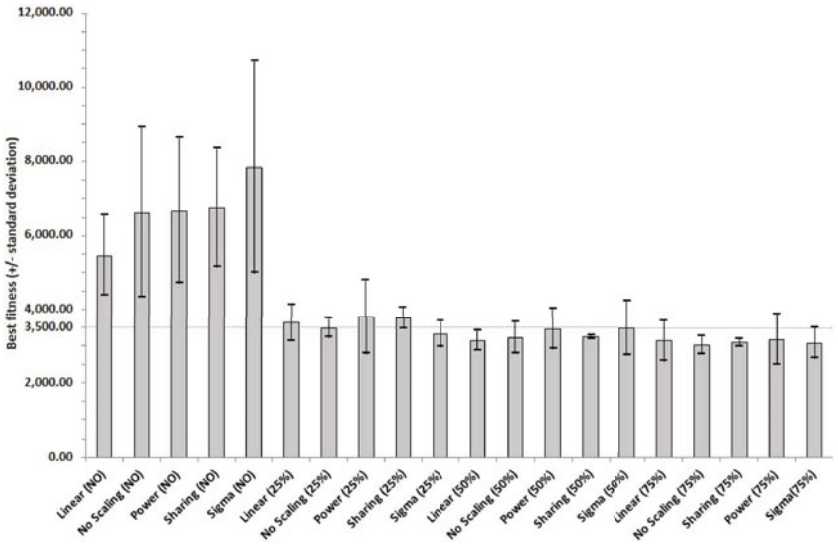
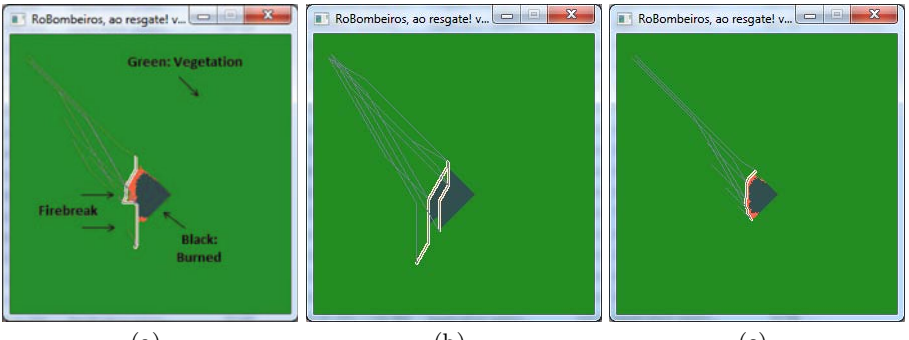


Fig. 1. Results of the evaluations presented in Table 2. The x-axis shows the description of the experiment as (scaling method; percentage of overlapping).

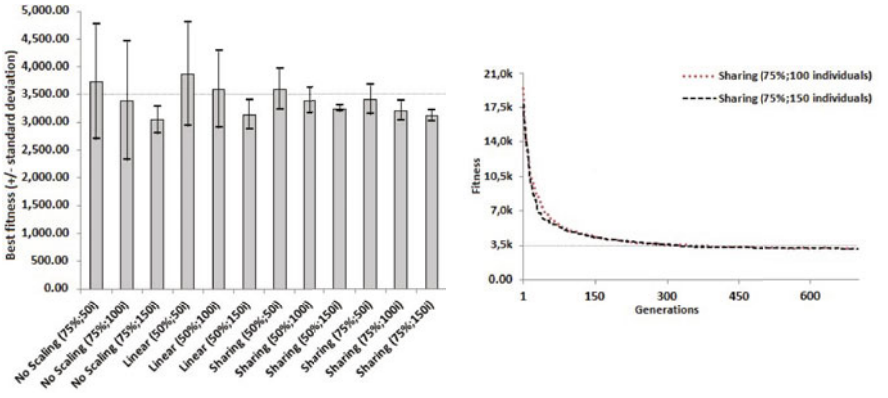


(a) (b) (c)

Fig. 2. (a) and (b) Unsatisfactory results. (a) Non overlapping with sigma scaling. (b) Non overlapping with no scaling. The firebreak are too large relative to what wold be necessary and are not well distributed among the robots. (c) Satisfactory result using sharing scaling and 75% of overlapping.

any user of GA [15]. If the population size is too small, the GA may converge too fast (with unsatisfactory result). If the population is too large, there may be a waste of computational resources.

We can see in Fig. 3(a) that decreasing in the number of individuals, the simulations results are worse. However, even using 100 individual instead of 150, one parameter set could obtain all results with fitness below 3,500 units. This parameter set is that used 75% of overlapping and sharing scaling. No other set



(a) (b)

Fig. 3. (a) Results of the evaluations considering the four best sets from Fig. 1 but using different populations size. The x-axis shows the parameter set as (scaling method; percentage of overlapping; number of individuals (b) Evolution of fitness according to number of generations, with variable population size.

(a) (b)

Fig. 4. (a) Four robots creating a firebreak. (b) Detailed view of mobile robots moving to the fire.

could obtain all simulations below 3,500 using 100 individuals. Fig. 3(b) present the fitness curve considering the best set described on previous paragraph. We can see that using both 100 or 150 individuals, the curves are quite similar.

Fire spread simulation considered East-West wind direction and relative wind speed at 7km/h; robot navigation speed of 35km/h; robots positioned on 2km far from fire threshold base. The simulations results show that, for the proposed fitness function and chromosome structure, the use of 75% of percentage of overlapping and sharing scaling allowed to obtain best results. Fig. 4 presents satisfactory evolution result applied on 3D virtual simulation environment. The 3D prototype showed that robots completely surround the fire and create the firebreaks on a satisfactory way.

6 Conclusions and Future Work

In this paper we evaluate the use of overlapping populations and fitness scaling in a GA applied to multi-robot squad formation and coordination. The robotic task is performed over a natural disaster scenario (a forest fire simulation). The robot squad mission is surrounding the fire and avoiding fire's propagation based on the strategy proposed by the GA. Simulations have been carried out to evaluate the effect of fitness scaling and overlapping population in the efficiency of GA search. The simulation's results shows that the use of overlapping population and fitness scaling presents better results than non-overlapping population and unscaled fitness.

There are three future work planned: (i) adaptations to the model to allow experiments using real robots; (ii) the theoretical mathematical study about the influences of fitness scaling using the proposed chromosome; (iii) evaluations of scalability and flexibility of the proposed group formation model.

Acknowledgements

The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology – Critical Embedded Systems – Brazil), processes 573963/2008-8 and 08/57870-9. Also acknowledge CAPES and FAPESP for their financial support of this research (doctoral grant).

References

1. Arai, T., Pagello, E., Parker, L.: Guest editorial: advances in multirobot systems. IEEE Transactions on Robotics and Automation 18(5), 655–661 (2002)
2. Balch, T., Arkin, R.C.: Behavior-based formation control for multi-robot teams. IEEE Transactions on Robotics and Automation 14(6), 926–939 (1998)
3. Bekey, G.A.: Autonomous Robots: From Biological Inspiration to Implementation and Control. The MIT Press, Cambridge (2005)
4. Darpa: Darpa grand challenge webpage (2007), http://www.darpa.mil/grandchallenge
5. DeJong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis (1975)
6. Dorigo, M., Trianni, V., Sahin, E., Gro, R., Labella, T., Baldassarre, G., Nolfi, S., Deneubourg, J., Mondada, F., Floreano, D., Gambardella, L.: Evolving self-organizing behaviors for a swarm-bot. Autonomous Robots 17, 223–245 (2004)
7. Go, J., Browning, B., Veloso, M.: Accurate and flexible simulation for dynamic, vision-centric robots. In: Proceedings of 3th International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2004 (2004)
8. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Professional, Reading (1989)
9. Heinen, M.R., Osório, F.S.: Applying genetic algorithms to control gait of simulated robots. In: Electronics, Robotics and Automotive Mechanics Conference (2007)
10. Heinen, M., Osório, F., Heinen, F., Kelber, C.: Seva3d: Using artificial neural networks to autonomous vehicle parking control. In: International Joint Conference on Neural Networks, IJCNN 2006 (2006)

11. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
12. Hopgood, A.A., Mierzejewska, A.: Transform ranking: a new method of fitness scaling in genetic algorithms. In: Proc. AI 2008, Research and Development in Intelligent Systems XXV (2008)
13. IdMind: Projecto raposa (2009), http://raposa.idmind.pt
14. Macedo, A.R.M., Macedo, A.R.L., Duarte, J.B.F.: Robótica aplicada ao combate a incidentes. Revista TN Petróleo (53), 108–113 (2007)
15. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1996)
16. Mitchell, M.: An Introduction to Genetic Algorithms. The MIT Press, Cambridge (1996)
17. Mondada, F., Gambardella, L.M., Floreano, D., Nolfi, S., Deneubourg, J., Dorigo, M.: The cooperation of swarm-bots: physical interactions in collective robotics. IEEE Robotics & Automation Magazine 12(2), 21–28 (2005)
18. Nguyen, V.B., Morris, A.S.: Genetic algorithm tuned fuzzy logic controller for a robot arm with two-link flexibility and two-joint elasticity. Journal of Intelligent and Robotic Systems 49(1), 3–18 (2007)
19. Parker, L.E.: Distributed intelligence: Overview of the field and its application in multi-robot systems. Journal of Physical Agents 2(1), 5–14 (2008)
20. Pessin, G.: Evolução de Estratégias e Controle Inteligente em Sistemas Multi-Robóticos Robustos. Master's thesis, Universidade do Vale do Rio dos Sinos (2008)
21. Pessin, G., Osório, F.: Avaliação de um modelo tolerante a falhas para atuação tática de grupos robóticos utilizando algoritmos genéticos. In: XXXV Conferencia Latinoamericana de Informática (CLEI 2009), pp. 1–10 (2009)
22. Wall, M.: Galib – a c++ library of genetic algorithm components. Massachusetts Institute of Technology (1996)
23. Whitley, D.: The genitor algorithm and selection pressure: why rank-based allocation of reproductive trials is best source. In: Proceedings of the Third International Conference on Genetic Algorithms, pp. 116–121 (1989)
24. Yamaguchi, H.: Adaptive formation control for distributed autonomous mobile robotgroups. In: IEEE Int. Conf. on Robotics and Automation (1997)
25. Zhao, Y., Collins, E.G.: Robust automatic parallel parking in tight spaces via fuzzy logic. Robotics and Autonomous Systems 51, 111–127 (2005)
26. Zhou, Y.: An Area Exploration Strategy Evolved by Genetic Algorithm. Master's thesis, University of Georgia (2005)

VarMOPSO: Multi-Objective Particle Swarm Optimization with Variable Population Size

Javier López, Laura Lanzarini, and Armando De Giusti

III-LIDI(Instituto de Investigación en Informática LIDI)
School of Computer Sciences. National University of La Plata
La Plata, Buenos Aires, Argentina

Abstract. The PSO (Particle Swarm Optimization) metaheuristics, originally defined for solving single-objective problems, has been applied to multi-objective problems with very good results. In its initial conception, the algorithm has a fixed-size population. In this paper, a new variation of this metaheuristics, called VarMOPSO (Variable Multi-Objective Particle Swarm Optimization), characterized by a variable-sized population, is proposed. To this end, the concepts of age and neighborhood are incorporated to be able to modify the size of the population for the different generations. This algorithm was compared with the version that uses fixed-size populations, as well as with other metaheuristics, all of them representative of the state of the art in multi-objective optimization. In all cases, three widely used metrics were considered as quality indicators for Pareto front. The results obtained were satisfactory.

Keywords: Evolutionary Computation, Swarm Intelligence, Particle Swarm Optimization, Multi-objective Function Optimization.

1 Introduction

Problem optimization is a very frequent problem in real life. We are constantly faced with the need of finding solutions that maximize the performance or minimize the consumption of any given scarce resource. There is also a set of real-life problems, known as multiobjective problems, that have more than one objective to maximize or minimize simultaneously. Usually, there is a trade off between these objectives, and a set of compromise solutions to be used during the decision-making process is required [1].

Particle Swarm Optimization (PSO) is a metaheuristic proposed by Kennedy and Eberhart [2], and there are more than 30 variations that allow applying it to multi-objective problems. These variations are generally referred to as MOPSO (Multi-objective Particle Swarm Optimization). Among these, SMPSO and OMOPSO show the best perfomance [3].

In this paper, a new variation of a MOPSO-type algorithm, called VarMOPSO, incorporating the concept of variable population, is proposed. The same as the solution adopted in [5], this algorithm uses an external archive to maintain non-dominated solutions. To calculate particle speed, the velocity constriction equation proposed in [6], based on the constriction factor in [7], is used. Population

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 60–69, 2010.

variation between generations is the result of adding the concept of age to the particles and using an insertion/removal procedure based on the ability of each individual to solve the problem posed. The concept of elitism is used, and only dominated solutions can be removed. These procedures are based on the mono-objective PSO version presented in [8].

It is known that, for PSO, the selection of the leader is crucial when calculating the speed of the particles [9] [15] [17]. The method proposed here uses a different selection mechanism between the two phases of the algorithm. During the first part of the search, two leaders are selected at random from the external, non-dominated solutions archive, and a binary tournament process based on their crowding value is carried out, in accordance to the mechanism proposed in [5]. During the final part of the process, the previously described mechanism is used 50% of the times, whereas the remaining 50%, the leader with the less crowding value is deterministically selected in order to find new solutions in the less populated areas of Pareto front.

To assess the performance of the method proposed, comparisons with NSGAII [10], SPEA2 [11], and SMPSO [6] algorithms were carried out; the three are representative of the state of the art in multi-objective optimization. To carry out the experiments and obtain the results, the jMetal framework implementation [12] of these algorithms was used, with the added implementation of the method proposed.

This paper is organized as follows: in Section 2, the main concepts related to multi-objective optimization and the main aspects of the PSO metaheuristic are reviewed, a well as the extensions developed for multi-objective optimization. In Section 3, the proposed algorithm is described in detail. Then, in Section 4, detailed information is provided regarding the test functions used and the configuration of each algorithm. Finally, in Section 5, the conclusions and future lines of work are presented.

2 Multi-objective Optimization with PSO

A multi-objective problem implies the simultaneous optimization of several objective functions which are generally at conflict. In this type of problems, the solution consists in a set of optimum solutions. The reason for creating a set of solutions is that no single solution can be considered to be better than the rest when considering all the objective functions, since the best solution for one of the objective functions can be worse for another function.

In this context, Pareto's concept of dominance is used. Given a population S, the set of optimum solutions is defined as the subset of S that is formed by all individuals of S that are non-dominated. In the solutions space, it is known as Pareto set, and its image in the space of the objective functions is the Pareto front.

The main purpose of a multi-objective optimization algorithm is finding candidate solutions whose image is as close to the real Pareto front as possible, are evenly distributed and maximize its coverage [14].

PSO is a population heuristics technique where each individual represents a possible solution to the problem and adapts following three factors: its knowledge of the environment (its fitness value), its historical knowledge or previous experiences (its memory), and the historical knowledge or previous experiences of the individuals in its neighborhood (leader) [2].

The PSO metaheuristic has also been used to solve multi-objective problems with good results [6] [3] [13].

In an MOPSO, the selection of the leader greatly influences both the convergence of the algorithm with Pareto front and the diversity of the solutions found [9]. Elitism is necessary to converge to the Pareto Optimal Front [19]. In most MOPSO implementations, the non-dominated solutions file is a fixed-size external archive. There are different methods to select the non-dominated solutions to be removed when their number exceeds the preset size [3] [4] [9] [19]. When the execution of the algorithm ends, this external archive contains the set of solutions that form the Pareto front obtained. Ideally, this archive is a subset of solutions of the real Pareto front for the optimized problem.

3 varMOPSO: Variable Multi-Objetive Particle Swarm Optimization

The main contribution of our proposal is focused on the variation of the size of the population. This allows the algorithm to adapt the number of individuals to use in both phases, exploration and exploitation, based on the difficulty of the problem. During the exploration phase, which is the first phase of the algorithm, corresponding to the search of promising regions of the parameter space, the population grows until reaching 6 or 7 times the size of the initial population, depending on the problem and the execution. Then, as iterations go by, this number of individuals is gradually reduced until a minimum preset number is reached, which marks the exploitation phase of the algorithm. In the following paragraphs, the concepts and procedures incorporated to the algorithm to change the population size are described.

3.1 Life Time

The particle's life time is one of the main concepts in this work. It defines the permanence of the particle within the population. Such value is expressed in quantity of iterations; once it is over, the particle is removed unless it belongs to the set of non-dominated solutions for the population. This value is closely related to the capacity of each particle and allows the best to remain longer in the population, influencing the behavior of the others.

In order to assess the life time of each population individual, the individuals of a population are grouped according to their rank value in k classes. The rank value indicates the non-domination level of the individual in the population [10]. With the result of this grouping, the method proposed in [8] is applied. For this procedure, elitism is used. The solution is removed from the population only if

it is not a non-dominated solution. This procedure is executed while population size is greater than a minimum, preset value.

3.2 Particle Insertion

The particle insertion procedure compensates the particle elimination, increases the search capacity and the convergence speed. The quantity of incorporated particles at each iteration coincides with the quantity of isolated individuals. An isolated individual is an individual that does not have any neighbor within a calculated r radius, following the criterion adopted in [8].

$$d_i = min\{||x_i – x_j||; \forall j \quad x_i, x_j \in S; x_i \neq x_j\} \; i = 1..n \quad ; \quad r = \sum_{i=1}^{n} d_i/n \qquad (1)$$

The adopted criterion to determine the position of these new individuals was the following: 20% of these new particles receive a position vector of one of the best individuals of the population, but their speed vector is random; the remaining 80% are random, as assessed in [8]. To select the best individual of the population, two particles are randomly selected from the external archive and a binary tournament is carried out. We choose the solution that was located in a less-crowded region within the objective function space [10].

To prevent population size to grow excessively, there is a variable c that regulates population growth. This variable reflects the probability of creating a new particle. At the beginning of the algorithm, the value of this variable is 1.0. If the size of the population grows more than 5% between generations, the value of variable c is decreased by 0.1. When the variable reaches a value of 0, it keeps that value for the remainder of the execution. Thus, the algorithm increases the size of the population during the first iterations, reaching a maximum value. Once the variable c reaches a value of 0, the size of the population decreases gradually until reaching the minimum preset number of individuals.

3.3 Particle Removal

This procedure has the purpose of removing those particles that are in overpopulated areas within the parameter space.

First, the number of nearby neighbors for each particle is calculated. This number of nearby neighbors corresponds to the number of particles that surround an individual within a radius r_1, as described by the following equation:

$$r_1 = \sum_{i=1}^{n} (r - ||x_i – x_j||)/n \quad ; \quad \forall j \quad x_i, x_j \in S \; ; \; x_i \neq x_j \quad i = 1..n \qquad (2)$$

All particles with more than 0 nearby neighbors are candidates for removal.

For each individual, an aptitude value normalized between 0 and 1 is calculated considering two factors – the aptitude of the particle to solve the problem and the density of individuals surrounding each solution.

The aptitude value ap of each particle is calculated with the following equation.

$$a_i = (max(ra) – ra_i)/(max(ra) - min(ra)) \quad ; \quad p_i = (pr_i/max(pr))$$

$$ap_i = 0.5 * a_i + 0.5 * p_i \quad ; \quad i = 1..n \tag{3}$$

where

- $max(ra)$ = maximum dominance level in the subpopulation of particles that are candidates for removal;
- ra_i = dominance level of each particle;
- $max(pr)$ = maximum number of neighbors in the subpopulation of particles that are candidates for removal;
- pr_i = number of nearby neighbors of each particle.

The previous calculation always returns a value between 0 and 1. As it can be inferred, the aptitude value depends 50% on the ranking of the particle (non-domination level within the population) and 50% on the number of neighbors within the previously described radius r_1.

Then, for each particle, a random value between 0 and 0.50 is selected. If the aptitude value is lower than this value, the particle is removed; otherwise, it remains in the population. The probability of removal of any given particle is inversely proportional to the calculated aptitude value.

For this procedure, elitism is used. The solution is removed from the population only if it is not a non-dominated solution. This procedure is executed while population size is greater than a minimum, preset value.

Figure 1 shows a detail of the pseudo-code for the proposed algorithm.

```
Pop = CreatePopulation(N) {Initial population is generated}
ComputeFitness(Pop);
ComputeRank(Pop) {computes the nondomination level of each particle}
ComputeLifeTimes(Pop);
AddArchiveLeaders (Ext);
ComputeRadius(Pop, cant, Eliminados) {first time Eliminados is empty}
while no end condition is reached do
   NewPop = CreateNewParticles(Pop, cant);
   ComputeLifeTimes(NewPop)
   Pop = Pop ∪ NewPop;
   ComputeSpeed(Pop);
   ComputeNewPosition(Pop);
   ComputeFitness(Pop);
   AddArchiveLeaders(Ext);
   ComputeRank(Pop);
   Deduct 1 to each particles life time
   Remove the particles with null life time
   ComputeRadius(Pop, cant, Eliminados);
   EliminCrow(Eliminados);
   ComputeLifeTimes(Pop);
end while
Output : Ext
```

Fig. 1. Pseudo-code for the proposed method

The *ComputeLifeTimes* process receives a complete swarm and only computes the life time corresponding to the particles that have null life time.

The *AddArchiveLeaders* adds non-dominated solutions to the external archive, and eliminates dominated solutions from it. If the external archive size is greater than a fixed value, this function uses the crowding distance of NSGA-II [10] to decide which particles should remain in it.

The radius computation, according to equation (1) and (2), is carried out within the *ComputeRadius* process, which receives the complete swarm as parameter and returns the quantity of new particles that should be inserted in the population. This module is the one in charge of avoiding the concentration of several particles in the same place of the search space; for this reason, it also returns, in *Eliminados*, the list of individuals that have really close neighbors. Such particles are eliminated in the *EliminCrow* module, based on their normalized fitness calculated according to (3).

4 Experiments

To assess the performance of varMOPSO, the ZDT (Zitzler-Deb-Thiele) [18] and DTLZ (Deb-Thiele-Laumanns-Zitzler) [20] sets of functions, configured with 2 objectives, were used. The experiments were carried out for the NSGA II, SPEA2, SMPSO and varMOPSO algorithms. The following indicators were calculated: additive unary epsilon indicator, spread, and hypervolume [1] [14] [16], to compare the quality of the Pareto fronts obtained. These indicators are representative to measure the general convergence to the real Pareto front, the coverage of the Pareto front obtained, and the uniformity in the distribution of solutions reached [14] [16]. To develop the algorithm, as well as to carry out the experiment and generate the results, jMetal Framework [12] was used, given its ease of use and its good documentation.

Table 1. Hipervolumen. Median and Interquartile

	NSGAII	SPEA2	varMOPSO	SMPSO
ZDT1	$6.59e-01_{3.8e-04}$	$6.60e-01_{3.7e-04}$	$6.62e-01_{3.3e-04}$	$6.62e-01_{1.1e-04}$
ZDT2	$3.26e-01_{2.7e-04}$	$3.26e-01_{6.7e-04}$	$3.29e-01_{2.5e-01}$	$3.29e-01_{1.6e-04}$
ZDT3	$5.15e-01_{2.0e-04}$	$5.14e-01_{4.9e-04}$	$5.16e-01_{3.0e-04}$	$5.15e-01_{6.1e-04}$
ZDT4	$6.54e-01_{6.5e-03}$	$6.51e-01_{7.4e-03}$	$6.62e-01_{4.0e-04}$	$6.61e-01_{2.4e-04}$
ZDT6	$3.89e-01_{1.9e-03}$	$3.79e-01_{4.6e-03}$	$4.01e-01_{1.2e-04}$	$4.01e-01_{1.2e-04}$
DTLZ1	$4.87e-01_{7.8e-03}$	$4.85e-01_{5.7e-03}$	$4.95e-01_{2.0e-04}$	$4.94e-01_{2.9e-04}$
DTLZ2	$2.11e-01_{3.1e-04}$	$2.12e-01_{1.8e-04}$	$2.12e-01_{2.6e-04}$	$2.12e-01_{1.7e-04}$
DTLZ3	$0.00e+00_{3.8e-02}$	$0.00e+00_{0.0e+00}$	$2.12e-01_{1.6e-04}$	$2.12e-01_{1.3e-01}$
DTLZ4	$2.09e-01_{2.1e-01}$	$2.10e-01_{2.1e-01}$	$2.10e-01_{2.6e-04}$	$2.10e-01_{1.2e-04}$
DTLZ5	$2.11e-01_{3.3e-04}$	$2.12e-01_{1.5e-04}$	$2.12e-01_{2.8e-04}$	$2.12e-01_{1.5e-04}$
DTLZ6	$1.73e-01_{3.9e-02}$	$8.12e-03_{1.2e-02}$	$2.12e-01_{2.1e-01}$	$2.12e-01_{6.9e-05}$
DTLZ7	$3.33e-01_{2.3e-04}$	$3.34e-01_{2.7e-04}$	$3.34e-01_{5.4e-05}$	$3.34e-01_{1.1e-04}$

Table 2. Statistical Significance for Hypervolume in ZDT Functions

	SPEA2					varMOPSO					SMPSO				
NSGAII	▽	–	▲	–	▲	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽
SPEA2						▽	▽	▽	▽	▽	▽	▽	▽	▽	▽
varMOPSO											–	–	▲	▲	–

Table 3. Statistical Significance for Hypervolume in DTLZ Functions

	SPEA2							varMOPSO							SMPSO						
NSGAII	–	▽	–	▽	▽	▲	▽	▽	▽	▽	▽	▽	–	▽	▽	▽	▽	▽	▽	▽	▽
SPEA2								▽	▽	▽	▽	▽	–	▽	▽	▽	▽	▽	▽	▽	▽
varMOPSO															▲	–	▲	▲	▲	▽	▲

Table 4. SPREAD. Median and IQR

	NSGAII	SPEA2	varMOPSO	SMPSO
ZDT1	$3.69e-01_{5.8e-02}$	$1.50e-01_{2.3e-02}$	$8.32e-02_{2.5e-02}$	$7.83e-02_{1.2e-02}$
ZDT2	$3.71e-01_{3.1e-02}$	$1.56e-01_{1.8e-02}$	$8.10e-02_{3.8e-01}$	$7.53e-02_{1.9e-02}$
ZDT3	$7.47e-01_{2.6e-02}$	$7.08e-01_{8.6e-03}$	$7.09e-01_{1.2e-02}$	$7.09e-01_{9.8e-03}$
ZDT4	$3.93e-01_{4.4e-02}$	$2.82e-01_{1.1e-01}$	$8.50e-02_{1.3e-02}$	$9.04e-02_{1.5e-02}$
ZDT6	$3.64e-01_{4.9e-02}$	$2.26e-01_{2.1e-02}$	$9.77e-02_{6.8e-01}$	$8.80e-02_{8.9e-01}$
DTLZ1	$3.93e-01_{5.7e-02}$	$1.84e-01_{7.0e-02}$	$6.82e-02_{1.3e-02}$	$7.29e-02_{2.3e-02}$
DTLZ2	$3.74e-01_{5.3e-02}$	$1.51e-01_{1.9e-02}$	$1.30e-01_{1.4e-02}$	$1.25e-01_{2.2e-02}$
DTLZ3	$8.95e-01_{1.5e-01}$	$1.05e+00_{1.6e-01}$	$1.13e-01_{2.5e-02}$	$1.41e-01_{7.7e-01}$
DTLZ4	$3.97e-01_{6.2e-01}$	$1.54e-01_{8.6e-01}$	$1.14e-01_{3.7e-02}$	$1.23e-01_{2.7e-02}$
DTLZ5	$3.80e-01_{4.1e-02}$	$1.52e-01_{2.0e-02}$	$1.30e-01_{2.6e-02}$	$1.28e-01_{1.2e-02}$
DTLZ6	$8.67e-01_{3.6e-01}$	$8.16e-01_{1.2e-01}$	$1.09e-01_{3.6e-01}$	$1.01e-01_{2.7e-02}$
DTLZ7	$6.31e-01_{2.3e-02}$	$5.47e-01_{1.2e-02}$	$5.19e-01_{1.3e-03}$	$5.19e-01_{2.4e-03}$

Table 5. Statistical Significance for Spread in ZDT Functions

	SPEA2					varMOPSO					SMPSO				
NSGAII	▽	▽	▽	▽	▽	▽	▽	▽	▽	–	▽	▽	▽	▽	▽
SPEA2						▽	▽	–	▽	–	▽	▽	–	▽	–
varMOPSO											–	–	–	–	–

Table 6. Statistical Significance for Spread in DTLZ Functions

	SPEA2							varMOPSO							SMPSO						
NSGAII	▽	▽	▲	▽	▽	–	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽
SPEA2								▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽
varMOPSO															–	–	▲	–	–	–	–

For the first three algorithms, a fixed population of 100 individuals was used. In the case of varMOPSO, the initial population included 100 particles. The size of the external archive was 100 individuals in all cases.

In NSGA II [10] and SPEA2 [11] SBX and polynomial mutation were used as operators for the crossover and mutation operators, respectively. The crossover probability is 0.9 and the mutation probability is 1/L, where L is the number of

Table 7. EPSILON. Median and IQR

	NSGAII	SPEA2	varMOPSO	SMPSO
ZDT1	$1.26e-02_{2.1e-03}$	$8.88e-03_{6.3e-04}$	$5.76e-03_{4.7e-04}$	$5.60e-03_{1.5e-04}$
ZDT2	$1.32e-02_{2.6e-03}$	$9.04e-03_{1.2e-03}$	$5.63e-03_{5.4e-01}$	$5.58e-03_{2.9e-04}$
ZDT3	$7.98e-03_{1.8e-03}$	$9.90e-03_{2.5e-03}$	$5.45e-03_{9.1e-04}$	$5.73e-03_{1.2e-03}$
ZDT4	$1.62e-02_{3.9e-03}$	$3.47e-02_{5.6e-02}$	$5.93e-03_{6.0e-04}$	$6.13e-03_{5.8e-04}$
ZDT6	$1.43e-02_{2.5e-03}$	$2.42e-02_{5.0e-03}$	$4.76e-03_{4.2e-04}$	$4.68e-03_{3.3e-04}$
DTLZ1	$7.69e-03_{2.3e-03}$	$6.14e-03_{2.8e-03}$	$2.91e-03_{2.1e-04}$	$3.03e-03_{1.8e-04}$
DTLZ2	$1.18e-02_{3.3e-03}$	$6.92e-03_{1.1e-03}$	$5.05e-03_{3.1e-04}$	$5.28e-03_{1.7e-04}$
DTLZ3	$9.34e-01_{1.5e+00}$	$2.36e+00_{1.2e+00}$	$5.34e-03_{3.6e-04}$	$5.51e-03_{7.0e-01}$
DTLZ4	$1.08e-02_{9.9e-01}$	$8.60e-03_{9.9e-01}$	$5.42e-03_{4.3e-04}$	$5.39e-03_{3.7e-04}$
DTLZ5	$1.06e-02_{2.1e-03}$	$7.68e-03_{1.3e-03}$	$5.06e-03_{3.8e-04}$	$5.22e-03_{3.1e-04}$
DTLZ6	$4.39e-02_{3.0e-02}$	$3.07e-01_{5.6e-02}$	$5.44e-03_{8.2e-01}$	$5.10e-03_{5.4e-04}$
DTLZ7	$1.03e-02_{2.1e-03}$	$9.52e-03_{2.1e-03}$	$5.03e-03_{4.0e-04}$	$5.09e-03_{4.7e-04}$

Table 8. Statistical Significance for Epsilon in ZDT Functions

	SPEA2					varMOPSO					SMPSO				
NSGAII	▽	▽	▲	▲	▲	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽
SPEA2						▽	▽	▽	▽	▽	▽	▽	▽	▽	▽
varMOPSO											▽	–	–	–	–

Table 9. Statistical Significance for Epsilon in DTLZ Functions

	SPEA2							varMOPSO							SMPSO						
NSGAII	▽	▽	▲	–	▽	▲	–	▽	▽	▽	▽	▽	–	▽	▽	▽	▽	▽	▽	▽	▽
SPEA2								▽	▽	▽	▽	▽	–	▽	▽	▽	▽	▽	▽	▽	▽
varMOPSO															–	▲	▲	–	▲	▽	–

decision variables. In SMPSO [6], polynomial mutation was used. The mutation probability is 1/L where L is the number of decision variables. In varMOPSO, a minimum population of 10 individuals and k=4 classes for calculating life time were used.

In every case, 25,000 function assessments were used, and each algorithm was executed 25 times for each problem.

Table 1 sows the results of the hypervolume indicator for all the algorithms in all the problems. The table includes median and interquartile range (IQR) values. The best results are highlighted in dark gray, and the second best results in light gray. Tables 2 and 3 indicate if the results are statistically significant for the two groups of test functions.

In each table, a ▽ symbol implies a p-value < 0.05, indicating that the null hypothesis (both distributions have the same median) is rejected; otherwise, a ▲ or − are used. Similarly, Tables 4, 5 and 6 show the same information for the Spread indicator, and Tables 7, 8 and 9 show the Median and interquartile range of the (additive) Epsilon (I_ε) indicator.

5 Conclusions and Future Work

After an analysis of the results shown in the previous tables, the following conclusions can be drawn:

- Both MOPSO algorithms (SMPSO and varMOPSO) are capable of generating better-quality Pareto fronts, considering the three metrics used and the set of selected tests, than the NSGA II and SPEA2 algorithms.
- No significant differences are found between the performance of the SMPSO and varMOPSO algorithms. In some functions and indicators, the first algorithm presents the best results, whereas in others, the second algorithm offers a better performance; although the differences are not statistically significant considering a p-value < 0.05 (except for the Hypervolume and Epsilon indicators in the DTLZ6 function, where SMPSO shows better results).
- In general, varMOPSO shows a greater dispersion than SMPSO, but most of the time the former got the best individual result for the evaluated metrics in the different test functions.

Based on the discussion presented in this paper, we consider that our proposal is an interesting option for solving multi-objective problems. The method is based on an extension of the varPSO algorithm for multi-objective problems, incorporating improvements studied in state-of-the-art MOPSO algorithms.

The use of variable population algorithms is a rather unexplored alternative for multi-objective optimization. By changing the size of the population, a better adaptation to the problem at hand is achieved, and the possibilities of finding a balance between exploration and exploitation increase, which is desirable in any optimization algorithm. We believe that there is still much that can be done in the field of variable population MOEAs (multiobjective evolutionary algorithms).

References

1. Zitzler, E., Laumanns, M., Bleuler, S.: A Tutorial on Evolutionary Multiobjective Optimization, Swiss Federal Institute of Technology (ETH) Zurich. In: GECCO 2009 (2009)
2. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, vol. IV, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
3. Durillo, J.J., García Nieto, J., Nebro, A.J., Coello Coello, C.A., Luna, F., Alba, E.: Multi-objective particle swarm optimizers: An experimental comparison. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 495–509. Springer, Heidelberg (2009)
4. Raquel, C., Naval, P.: An effective use of crowding distance in multiobjective particle swarm optimization. In: Beyer, H. (ed.) 2005 Conference on Genetic and Evolutionary Computation, GECCO 2005, pp. 257–264. ACM, New York (2005)
5. Sierra, R., Coello, C.: Improving PSO-Based Multiobjective Optimization Using Crowding, Mutation and *epsilon*-Dominance. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 505–519. Springer, Heidelberg (2005)

6. Nebro, A.J., Durillo, J.J., García-Nieto, J., Coello Coello, C.A., Luna, F., Alba, E.: SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization. In: IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MCDM 2009), pp. 66–73 (March 2009)
7. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002)
8. Lanzarini, L., Leza, V., De Giusti, A.: Particle Swarm Optimization with Variable Population Size. In: 9th International Conference on Artificial Intelligence and Soft Computing Zakopane, Poland, pp. 438–449 (2008) ISBN: 978-3-540-69572-1
9. Sierra, R., Coello, C.: Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. International Journal of Computational Intelligence Research 2(3), 287–308 (2006)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
11. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Giannakoglou, K., et al. (eds.) EUROGEN 2001, Athens, Greece, pp. 95–100 (2001)
12. Durillo, J.J., Nebro, A.J., Luna, F., Dorronsoro, B., Alba, E.: jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics. Technical Report ITI-2006-10, Depto.de Lenguajes y Ciencias de la Computación, University of Málaga, E.T.S.I. Informática, Campus de Teatinos (December 2006)
13. Abido: Two-Level of Nondominated Solutions Approach to Multiobjective Particle Swarm Optimization. In: Genetic And Evolutionary Computation Conference Proceedings, pp. 726–733 (2007) ISBN:978-1-59593-697-4
14. Zitzler: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD Thesis, Swiss Federal Institute of Technology (ETH) Zurich (November 1999)
15. Shi, E.: Parameter Selection in Particle Swarm Optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 591–600. Springer, Heidelberg (1998)
16. Knowles, J.D., Tiele, S., Zitzler, E.: A Tutorial on the Perfomance Assessment of Stochastic Multiobjective Optimizers, TIK - Report 214, ETH Zurich (2006)
17. Van den Bergh: An Analysis of Particle Swarm Optimizers. Ph.D. dissertation. Department Computer Science. University Pretoria, South Africa (2002)
18. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8(2), 173–195 (2000)
19. Mostaghim, S., Teich, J.: Covering Pareto optimal Fronts by Subswarms in Multiobjective Particle Swarm Optimization. In: Congress on Evolutionary Computation (2), 1404 (2004)
20. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. In: Theoretical Advances and Applications, pp. 105–145. Springer, Heidelberg (2001)

A Multi-Objective Genetic Algorithm with Path Relinking for the p-Median Problem

José E.C. Arroyo, Paula M. dos Santos, Michele S. Soares, and André G. Santos

Departamento de Informática, Universidade Federal de Viçosa
Campus Universitário da UFV, 36570-00 Centro Viçosa, MG, Brazil
{jarroyo,andre}@dpi.ufv.br, paula-marianna@hotmail.com,
myxellys@gmail.com

Abstract. This paper considers the *p*-median problem that consists in finding *p*-locals from a set of *m* candidate locals to install facilities minimizing simultaneously two functions: the sum of the distances from each customer to its nearest facility and the sum of costs for opening facilities. Since this is a NP-Hard problem, heuristic algorithms are the most suitable for solving such a problem. To determine nondominated solutions, we propose a multi-objective genetic algorithm (MOGA) based on a nondominated sorting approach. The algorithm uses an efficient elitism strategy and an intensification operator based on the Path Relinking technique. To test the performance of the proposed MOGA, we develop a Mathematical Programming Algorithm, called ε-Constraint, that finds Pareto-optimal solutions by solving iteratively the mathematical model of the problem with additional constraints. The results show that the proposed approach is able to generate good approximations to the nondominated frontier of the bi-objective problem efficiently.

Keywords: Genetic Algorithms, Multi-objective Combinatorial Optimization, p-Medians, Path Relinking.

1 Introduction

Given *n* customers and *m* potential facilities, the single-objective *p*-median problem (SO-*p*-MP) is the most well-known facility location problem. It consists in finding a subset with *p* facilities (medians) such that the total sum of distances between each customer and its nearest facility is minimized. In the non-capacitated SO-*p*-MP, one considers that each facility candidate to median can satisfy an unlimited number of customers. Facility location-allocation problems have several applications in telecommunications, industrial transportation and distribution, clustering, etc.

The SO-*p*-MP is NP-hard [15]. Exact and heuristic methods have been proposed for solve this problem. Exact algorithms are provided by [3], [4] and [9]. Greedy heuristics were proposed by [16] and [23]. In [22], [13] and [19] were presented local search heuristics. Modern heuristics (metaheuristics) have been applied to the SO-*p*-MP as well [18]. Tabu Search (TS) algorithms were designed by [20] and [21]. Simulated Annealing (SA) heuristics were proposed by [6] and [17]. In [1], [5] and

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 70–79, 2010.

[14] were suggested GAs. Variable Neighborhood Search heuristics have been proposed by [7] and [13]. GRASP (greedy randomized adaptive search procedure) heuristic was designed by [19].

In this paper, we examine the *p*-median problem from a bi-objective point of view. Two objectives are simultaneously minimized: the sum of the distances from each customer to its nearest facility (Z_1) and the total cost of opening the p facilities (Z_2). We use a multi-objective decision-making method to obtain a trade-off solution.

The presence of multiple objectives in an optimization problem, in principle, gives rise to a set of optimal solutions (known as *Pareto-optimal* or *efficient* solutions), instead of a single optimal solution. In the absence of any further information, none of these Pareto-optimal solutions can be said to be better than the other (the solutions are nondominated). Knowledge about the set of the *Pareto-optimal* solutions helps the decision maker in choosing the best compromise solution.

The most used methods for solving multi-objective optimization problems are heuristics and metaheuristic [11]. Metaheuristic methods were originally conceived for single-objective combinatorial optimization and the success achieved in their application to a very large number of problems has stimulated researchers to extend them to multi-objective combinatorial optimization problems. Surveys of multi-objective metaheuristics research reveal that in most articles from the literature use GAs as the primary metaheuristic, followed by SA and TS [11]. Since GAs work with a population of solutions, it seems natural to use GAs in multi-objective optimization problems to capture a number of solutions simultaneously [10].

The purpose of this paper is to test an effective intensification procedure based on the Path Relinking technique for improving the performance of a MOGA. The proposed MOGA uses the fast nondominated sorting approach developed by [10] in which the selection process is based on classifying the population in several fronts of nondominated solutions. The Path Relinking technique combines a solution generated by crossover/mutation operators with a nondominated solution found by the algorithm. We also test an efficient elitism strategy that consists in adding in the population a number of nondominanted solutions [2]. These elite solutions will participate of the selection and crossover operators.

To evaluate the performance of the proposed MOGA, we implemented a bi-objective Mathematical Programming algorithm, called ε-Constraint, that determines Pareto-optimal solutions by solving the Integer Programming model of the SO-*p*-MP. This algorithm iteratively minimizes an objective (Z_1 or Z_2) considering different upper bounds for the other objective.

2 Formulation of the Bi-Objective *p*-Median Problem (BO-*p*-MP)

Consider a set J of m location points (for installing facilities), a set I of n users (or customers or demand points), a $n \times m$ matrix with the distances traveled d_{ij} for satisfying the demand of the user located at i from the facility located at j, for all $j \in J$ and $i \in I$ and the fixed cost f_j of operating local j (the cost of installing or opening a facility at the local j). The BO-*p*-MP consists in finding a subset of L with p facilities (medians) in order to minimize, simultaneously two objectives, Z_1: the sum of the distances from each customer to its nearest facility and Z_2: the total cost of opening the p facilities. The Integer Programming formulation of the BO-*p*-MP is as follows:

$$\text{Minimize } Z_1 = \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \tag{1}$$

$$\text{Minimize } Z_2 = \sum_{j \in J} f_j y_j \tag{2}$$

(BO-*p*-MP)

$$\text{subject to: } \sum_{j \in J} x_{ij} = 1, \quad i \in I \tag{3}$$

$$x_{ij} \leq y_j, \quad i \in I, j \in J \tag{4}$$

$$\sum_{j \in J} y_j = p \tag{5}$$

$$x_{ij}, y_j \in \{0,1\}, \quad i \in I, j \in J. \tag{6}$$

The decision variables of the problem are y_i and x_{ij}. $y_j = 1$, if a facility is opened in local $j \in J$, and 0, otherwise; $x_{ij} = 1$, if customer $i \in I$ is served from the facility located in $j \in J$, and 0, otherwise. The objective functions to be minimized are defined in (1) and (2). Constraints (3) guarantee that each customer is attended by one and only one facility. Constraints (4) prevent any customer from being supplied from a local with an unopened facility. The total number of open facilities is set to p by constraint (5). Finally, (6) defines that the decision variables are binary.

A solution s of the BO-*p*-MP belongs to *space of feasible solutions*, denoted for Ω, if it satisfies the Constraints (3) to (6). *Feasible objective space* of this problem is defined by $Z = \{q \in R^2: q = (Z_1(s), Z_2(s)), \forall s \in \Omega\}$. A solution s when mapped into the objective space is a point $q = (Z_1(s), Z_2(s))$.

The solutions of a bi-objective optimization problem are characterized by the following definitions (considers, without loss of generality, the minimization of the two objectives):

Definition 1 (*concept of dominance*):
Solution s_1 *dominates* s_2 if the following three conditions are satisfied:

i) $Z_1(s_1) \leq Z_1(s_2)$, ii) $Z_2(s_1) \leq Z_2(s_2)$ and iii) $Z_1(s_1) < Z_1(s_2)$ or $Z_2(s_1) < Z_2(s_2)$.

Similarly, a point $q_1 = (Z_1(s_2), Z_2(s_2))$ dominates $q_2 = (Z_1(s_2), Z_2(s_2))$ if the same three conditions are satisfied.

Definition 2 (*Pareto-optimal solutions*):
A solution s is *Pareto-optimal* or *efficient* solution if there is no $s' \in \Omega$ such that s' dominates s.

In the objective space, the *Pareto-optimal* points define the *Pareto-frontier*. Therefore, a bi-objective optimization problem consists in determine the set of *Pareto-optimal* solutions.

3 Multi-Objective Genetic Algorithm with Path Relinking

In the present work, two mechanisms were added to a multi-objective genetic algorithm (MOGA) to improve its searching ability: an effective elitism strategy and an intensification mechanism based on the Path Relinking technique.

The selection operator of the proposed MOGA is based on nondomination ranks (nondominated sorting) and *crowded* distances (sharing function) [10].

Algorithm MOGA+PR (N, N_{elite} , p_R, p_M, N_{Iter})
1: Initialize the set of nondominated solutions: $D = \varnothing$
2: Initialize the set of elite solutions: $E = \varnothing$
3: Generate a population P of size N
4: **For** i = 1 to N_{Iter} **do**:
5: Rank the population $P \cup E$ and obtain K fronts F_1, F_2,..., F_K
6: D = nondominated solutions of $(D \cup F_1)$ *//Updating the nondominated set*
7: *Update the set of elite solutions*:
7.1: If $| D | > N_{elite}$, then set $n_e = N_{elite}$, otherwise set $n_e = | D |$
7.2: E = a new set of n_e solutions randomly selected from D.
8: For each $x \in F_i$ (i = 1, 2, ..., K) calculate *crowded* distances: $dist(x)$
9: Use the usual *binary tournament selection* to choose N solutions from $P \cup E$
10: Use *crossover* and *mutation* operators to create an offspring population P' of size N
11: For each solution in P', compute the value of the objectives
12: *Intensification:*
12.1: s_o = Select randomly a solution from P'
12.2: s_g = Select randomly a nondominanted solution from D
12.3: L_{PR} = **Path-Relinking**(s_o, g_g); *// L_{PR} : the returned set of nondominated solutions*
12.4: D = nondominated solutions of $(D \cup L_{PR})$
14: $P = P'$
15: **End-for**
Return D
End-Algorithm

Fig. 1. Pseudocode of the proposed Multi-objective Genetic Algorithm

The elitism mechanism is similar to that used in [2]. A set E with N_{elite} solutions is selected from the set of the current nondominated solutions D and included in the population P, i.e., at the next iteration we have the combined population $P \cup E$. The solutions of this population will participate of the selection operator.

The population $P \cup E$ is ranked into K sub-populations or fronts, F_1, F_2,...,F_K, according to the dominance relationship between the solutions. This rank can be obtained by a fast sorting algorithm proposed by [10]. The front F_1 corresponds to the set of nondominated solutions. The front F_i, i = 2,…,K contains the nondominated solutions when the fronts F_1,…, F_{i-1} are removed from the combined population. The lower nondomination rank of a solution, the better it is.

To make difference among the solutions that belong to the same front F_k, to each solution is assigned a distance measure called *crowded* (*dist*) that is calculated considering the objective values. For each solution, an estimate about the number of solutions localized around it is determined (i.e. the perimeter of the cuboid calculated between the two nearest neighbors). The solutions more dispersed of a F_k front will have the biggest values of *dist* and, so, the biggest "probabilities" of selection.

When comparing two solutions belonging to the same front, extreme solutions prevail over not extreme ones. If both solutions are not extreme, the one with the bigger *dist* wins. This way extreme solutions and less crowded areas are encouraged [10].

The pseudocode of the MOGA with Path Relinking (MOGA+PR) is summarized in Figure 1. In the next subsections, we describe the implementation of the initial population generation, crossover and mutation operators and the Path Relinking technique, when the MOGA+PR is applied to the bi-objective p-MP.

3.1 Generation of an Initial Population

The initial population of size N is formed by nondominated solutions obtained by a constructive algorithm and by solutions generated randomly. For the BO-p-MP, a solution s is interpreted as a set of p open facilities (or medians).

Initially we construct N_s ($N_s < N$) solutions using the *sample greedy* algorithm proposed by [19] for the SO-p-MP. From the N_s constructed solutions, the nondominated solutions are stored in a set L. To complete the population, $N - |L|$ are generated randomly. In the *sample greedy* algorithm, a solution is constructed trying to minimize the value of the weighted linear utility function $Z_\lambda = \lambda_1 Z_1 + \lambda_2 Z_2$, where λ_1 and λ_2 are nonnegative weights define randomly such that $\lambda_1 + \lambda_2 = 1$.

3.2 Crossover and Mutation Operators

Two selected parent solutions s_1 and s_2 ($s_1 \neq s_2$) are recombined to generate two solutions s_{c1} and s_{c2} (children). Initially, we determine the difference sets $s_1 - s_2$ and $s_2 - s_1$. For example, consider two parent solutions $s_1 = [1, 2, 3, 4, 5]$ and $s_2 = [2, 5, 9, 10, 12]$. The difference sets are: $s_1 - s_2 = \{1, 3, 4\}$ and $s_2 - s_1 = \{9, 10, 12\}$. The value $\Delta = |s_1 - s_2| = |s_2 - s_1|$ is known as the *symmetric difference* between s_1 and s_2. The solution s_{c1} is generated as follows: nt elements that belong to $s_1 - s_2$ are removed from s_1 and nt elements that belong to $s_2 - s_1$ are inserted in s_1. The solution s_{c2} is generated as follows: the nt elements inserted in s_1 are removed from s_2 and the nt elements removed from s_1 are inserted in s_2. nt is an integer number generated randomly in $[1, \Delta - 1]$. That means that, nt is the number of swaps to be done between the two parent solutions. For example, for $nt = 2$, from the solutions s_1 and s_2 defined above we can generate the solutions $s_{c1} = [\underline{\mathbf{9}}, 2, \underline{\mathbf{12}}, 4, 5]$ and $s_{c2} = [2, 5, \underline{\mathbf{1}}, 10, \underline{\mathbf{3}}]$, doing swaps of 1 by 9 and 3 by 12.

Mutation is performed as follows. A facility $a \in s$ is swapped by a facility $b \notin s$. The facilities a and b are chosen randomly. For example, the mutation applied to the solution $s_{c1} = [9, 2, \mathbf{12}, 4, 5]$ may generate the solution $s_{c1} = [9, 2, \mathbf{8}, 4, 5]$, which means that the opened facility 12 was swapped by the closed facility 8.

3.3 Intensification with Path Relinking

The Path Relinking (PR) technique was originally proposed by [12] as a strategy of intensification where the objective is to explore the search space or path between two elite solutions (high-quality solutions), the first is called *origin* and the second, *guide*. The origin solution s_o is transformed into a solution of s_g making a sequence of movements. Starting from s_o, a path is generated in the neighborhood space that leads toward the other solution s_g. This is accomplished by selecting moves that introduce attributes contained in the guiding solution. The "best" move is the one that reduces the distance from s_o to s_g. This move is executed, that is, the solution obtained by the best move is chosen. In each step, solutions less similar to the origin solution s_o and more similar to the guide one s_g are obtained.

In this work, the origin solution s_o is selected from the population P generated by the genetic operators and the guide solution s_g is selected from the current set D of nondominated solutions. The PR procedure starts with the solutions s_o and gradually

transforms it into the other solution s_g by inserting elements from $s_g - s_o$ and removing elements from $s_o - s_g$. That is, in each stage we make swap movements. The total number of swaps made (to transform s_o into s_g) is $\Delta = |s_g - s_o| = |s_o - s_g|$. In multi-objective case, the "best" solutions are those which are nondominated by the all swap neighbors of the current stage. From the set of nondominated neighbors, we choose randomly a solution s to be analyzed. In the next stage, new solutions are generated from this solution s. The PR ends when s and s_g become equal. The set of the overall non-dominated solutions (set L_{PR}) is kept as the result of this procedure.

4 The ε-Constraint Algorithm for the BO-*p*-MP

The ε-Constraint algorithm approximates the set of Pareto-optimal solutions (or *Pareto-frontier*) by using an iterative scheme. The algorithm has two phases (see Figure 2).

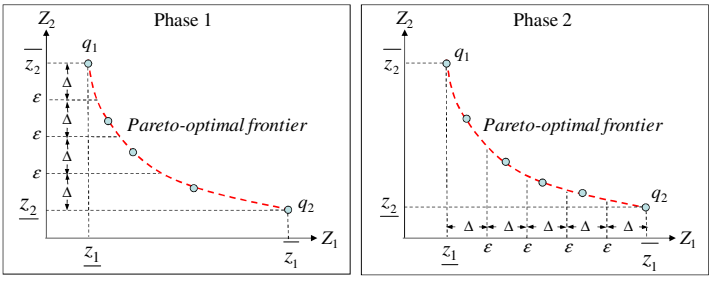


Fig. 2. ε-Constraint algorithm: Phases 1 and 2

Each phase determines a subset of Pareto-optimal solutions. The first phase starts by finding the extreme point $q_1 = (\underline{Z_1}, \overline{Z_2})$ of the Pareto-optimal frontier, where:

$\underline{Z_1}$ = *Min* Z_1 subject to: (3), (4), (5) and (6).

$\overline{Z_2}$ = *Min* Z_2 subject to: (3), (4), (5), (6) and $Z_1 \leq \underline{Z_1}$.

Then, the algorithm determines the minimum value of Z_2 ($\underline{Z_2}$), solving the model:

$\underline{Z_2}$ = *Min* Z_2 subject to: (3), (4), (5) and (6).

To find a set of *Pareto-optimal* points, the algorithm divides the range $\overline{Z_2} - \underline{Z_2}$ in s equally sized intervals. The size of each interval is determined by $\Delta = (\overline{Z_2} - \underline{Z_2})/s$.

The iterative scheme starts with the upper bound $\varepsilon = \overline{Z_2} - \Delta$ for Z_2 and a new *Pareto-optimal* point $q = (Z^*_1, Z^*_2)$ is determined solving the following models:

Z^*_1= *Min* Z_1 subject to: (3), (4), (5), (6) and $Z_2 \leq \varepsilon$

Z^*_2= *Min* Z_2 subject to: (3), (4), (5), (6) and $Z_1 \leq Z^*_1$.

Other points are determined considering the upper bounds $\varepsilon = \overline{Z_2} - K\Delta$, $K = 2,\ldots,s$. The Phase 1 of the algorithm finishes when $\varepsilon = \underline{Z_2}$.

The Phase 2 is a similar and complementary algorithm. It starts with the other extreme point $q_2 = (\overline{Z_1}, \underline{Z_2})$ of the Pareto-optimal frontier and new Pareto-optimal points $q = (Z^*_1, Z^*_2)$ are determined solving the following models:

$Z^*_2 = Min\ Z_2$ subject to: (3), (4), (5), (6) and $Z_1 \leq \varepsilon$
$Z^*_1 = Min\ Z_1$ subject to: (3), (4), (5), (6) and $Z_2 \leq Z^*_2$.

5 Computational Tests

The proposed algorithm MOGA+PR was coded in C++. The ε-Constraint algorithm was coded in Mosel and solved with Xpress-MP. The algorithms were executed on an Intel Core 2 Quad with a 2.4 GHz processor and 3GB of RAM. The computational tests were realized using a set of 150 test problems. In this work we considered problems in which the number of location points (m) is equal to the number of customers (n), that is, we generate square matrices with distances d_{ij}. The sizes of the problems are (m = 50, 100, 200, 300, 402)×(p = 5, 10, 15, 20). For each combination of m and p, we generated 5 problems based on single-objective p-MP instances available in the literature.

The MOGA algorithm presented in this study employs five parameters. The combination of parameters which yielded the best results are: N = 200 (size of the population), N_s = 100 (number of constructed solutions using the *sample greedy* algorithm), N_{elite} = 20 (maximum number of elite solutions), p_R= 0.9 (crossover probability), p_M = 0.3 (mutation probability) and N_{Iter} = 4000 (number of iterations).

The ε-Constraint algorithm use only the parameter s (number of intervals). For each phase of the algorithm, s was set to 50.

For each problem tested, we compare the solutions obtained by the algorithms ε-Constraint, MOGA (genetic algorithm without Path Relinking) and MOGA+PR (genetic algorithm with Path Relinking). We denoted by E the set of Pareto-optimal solutions obtained by ε-Constraint algorithm (Phases 1 and 2). D' and D are the sets of non-dominated solutions obtained by MOGA and MOGA+PR algorithms, respectively. We determine the reference set R of the nondominated solutions obtained by the three algorithms (R = nondominated solutions of $E \cup D' \cup D$). Note that, $E \subset R$, because set E contains only Pareto-optimal solutions.

For MOGA and MOGA+PR we compute the number of nondominated solutions in R: $|D' \cap R|$ and $|D \cap R|$ (cardinal measures).

We also measure the performance of the set X (X = E, D' or D) relative to the reference set R by using the following distance metric [8]:

$$D_{av} = 100 \times \frac{1}{|R|} \sum_{q \in R} \min_{q' \in X} d(q', q) \tag{7}$$

where, $q = (Z_1, Z_2) \in R$, $q' = (Z'_1, Z'_2) \in X$, $d(q, q') = \max\{(Z_1 - Z'_1)/\Delta_1\ ,\ (Z_2 - Z'_2)/\Delta_2\}$ and Δ_j (j =1,2) is the range of the objective Z_j in the reference set R.

Note that D_{av} is the average distance from a point $q \in R$ to its closest point in X. This distance metric evaluates the distribution of X as well as the proximity of X to R.

Table 1 presents the comparison among ε-Constraint, MOGA and MOGA+PR for problems with p = 5 and 10 medians. For each set of problems $m \times p$, Table 1 shows the total number of reference solutions provided by each algorithm (on five problems) and the average distance values (D_{av}) for each algorithm. For the 50 problems (with p=5 and 10), 2778 reference solutions were obtained, from which 2061 (74%) Pareto-optimal solutions were obtained by ε-Constraint algorithm, 2088 (75%) by MOGA and 2224 (80%) solutions were obtained by MOGA+PR algorithm. From Table 1, we can observe that the MOGA+PR algorithm finds a larger number of reference solutions than MOGA, and these solutions are closer to the reference solutions (see the values of D_{av}).

Table 1. Performance of the algorithms for problems with p = 5 and 10 medians

Problems	**Reference**	**ε-Constraint**		**AGMO**		**AGMO+PR**	
$m \times p$	$\|R\|$	$\|E\|$	D_{av}	$\|R \cap D'\|$	D_{av}	$\|R \cap D\|$	D_{av}
50×5	151	136	0.037	148	0.007	148	0.007
100×5	144	123	0.031	136	0.020	137	0.016
200×5	163	138	0.045	151	0.023	155	0.020
300×5	175	152	0.027	139	0.194	139	0.162
402×5	271	203	0.071	207	0.076	220	0.070
Total	***904***	***752***	***0.211***	***781***	***0.320***	***799***	***0.275***
50×10	286	215	0.063	260	0.177	275	0.048
100×10	287	229	0.054	206	0.597	225	0.525
200×10	385	290	0.060	231	1.399	281	0.482
300×10	455	293	0.071	317	0.594	325	0.418
402×10	461	282	0.072	293	0.842	319	0.602
Total	***1874***	***1309***	***0.320***	***1307***	***3.60***	***1425***	***2.075***

Table 2. Mean computational time (in seconds) for problems with p = 5 and 10 medians

Problems	***p*=5**			***p*=10**		
n	**ε-Constr.**	**MOGA**	**MOGA+PR**	**ε-Constr.**	**MOGA**	**MOGA+PR**
50	54.5	2.1	2.5	128.4	3.2	3.9
100	163.8	3.5	4.2	299.3	6.4	7.0
200	1662.1	6.7	7.3	2674.9	12.2	13.5
300	10086.1	10.0	10.6	13314.4	17.9	19.5
402	41362.7	13.1	14.2	54856.3	23.6	27.4
Total	***53329.2***	***35.4***	***38.8***	***71273.3***	***63.3***	***71.3***

Table 3. Performance of the algorithms MOGA and MOGA+PR for p =15, 20, 25 and 30

Problems	**Reference**	**MOGA**			**MOGA+*PR***		
p	$\|R\|$	$\|R \cap D'\|$	D_{av}	*Time*	$\|R \cap D\|$	D_{av}	*Time*
15	***2283***	***1881***	***1.306***	***86.3***	***2158***	***0.182***	***90.2***
20	***3071***	***2548***	***0.791***	***105.5***	***2927***	***0.106***	***111.6***
25	***3880***	***3185***	***0.830***	***134.4***	***3573***	***0.229***	***142.7***
30	***4691***	***3498***	***1.307***	***172.4***	***4513***	***0.040***	***181.8***

Table 2 shows the mean computational times (in seconds) spent by each algorithm for problems with p = 5 and 10. Note that the ε-Constraint algorithm spends more computational time. This algorithm, to solve a problem with 402×10, spends about 15 hours in average. For problems with $p \geq 15$, the computational time of the ε-Constraint algorithm is very expensive and it was impossible to generate Pareto-optimal solutions.

In Table 3, for each set of 25 problems with p =15, 20, 25 and 30, we present the results obtained by MOGA and MOGA+PR. In this table we show the total numbers of reference solutions, the total average distances and the total computational times. For the 100 problems (with p =15, 20, 25 and 30), 13925 reference solutions were obtained, from which 11125 (79.8%) and 13171 (94.5%) solutions were obtained by MOGA and MOGA+PR, respectively. For these problems, the MOGA+PR has found also a larger number of reference solutions. The computational times spent by MOGA+PR are slightly greater than the times of MOGA.

We also test the efficiency of the elitism strategy used in the MOGA. For it, we compare MOGA with elitism (MOGA+E) and MOGA without elitism (MOGA-E). For the 150 problems solved by MOGA-E and MOGA+E, 14642 reference solutions were obtained, from which 14516 (99%) solutions were obtained by MOGA+E, 3953 (27%) solutions were obtained by MOGA-E and 3827 identical solutions were generated by the two algorithms.

The MOGA with elitism and intensification strategies is able to determine a set of good nondominated solutions, very close to the Pareto-optimal solutions, of the bi-objective p-median problem.

6 Conclusions

In this paper we propose a MOGA with Path-Relinking to solve a bi-objective p-median problem. The goal of the proposed algorithm is to produce close to Pareto-optimal solutions. For problems with up to 402 location points and 10 medians, Pareto-optimal solutions were determined by a Mathematical Programming algorithm.

In this work we show that, the combination of the elitist MOGA and Path-Relinking provides an useful tool to solve multi-objective optimization problems. For a total of 150 bi-objective problems tested, 16703 reference (nondominated) solutions were obtained, from which 12878 (77%) solutions were obtained by MOGA and 15395 (92%) solutions were obtained by MOGA+PR (2517 more).

Acknowledgment. This work was funded by the CNPq and FAPEMIG. We also thank the Dash Optimization for providing full license of Xpress-MP.

References

1. Alp, O., Erkut, E., Drezner, Z.: An Efficient Genetic Algorithm for the p-Median problem. Annals of Operations Research 122, 21–42 (2003)
2. Arroyo, J.E.C., Armentano, V.A.: Genetic Local Search for Multi-Objective Flowshop Scheduling Problems. European Journal of Operational Research 167, 717–738 (2005)

3. Avella, P., Sassano, A.: On the p-Median Polytope. Mathematical Programming 89(3), 395–411 (2001)
4. Beasley, J.E.: A Note on solving Large p-median problems. European Journal of Operational Research 21, 270–273 (1985)
5. Chaudhry, S.S., He, S., Chaudhry, P.E.: Solving a class of facility location problems using genetic algorithm. Expert Systems 20, 86–91 (2003)
6. Chiyoshi, F., Galvão, R.D.: A Statistical Analysis of Simulated Annealing Applied to the p-Median Problem. Annals of Operations Research 96, 61–74 (2000)
7. Crainic, T., Gendreau, M., Hansen, P., Mladenovic, N.: Cooperative parallel variable neighborhood search for the p-median. Journal of Heuristics 10, 293–314 (2004)
8. Czyzak, P., Jaszkiewicz, A.: Pareto simulated annealing - a metaheuristic technique for multiple objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis, 734–747 (1998)
9. Daskin, M.S.: Network and Discrete Location: Models, Algorithms and Applications. Wiley, New York (1995)
10. Deb, K., Agrawal, S., Pratab, A., e Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
11. Gandibleux, X., Ehrgott, M.: 1984-2004 - 20 Years of Multiobjective Metaheuristics. But What About the Solution of Combinatorial Problems with Multiple Objectives? In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 33–46. Springer, Heidelberg (2005)
12. Glover, F.: Tabu search and adaptive memory programming - advances, applications, and challenges. In: Interfaces in Computer Science and Operations Research, pp. 1–75. Kluwer Academic Publishers, Dordrecht (1996)
13. Hansen, P., Mladenovic´, N.: Variable neighborhood search for the p-median. Location Science 5, 207–226 (1997)
14. Hosage, C.M., Goodchild, M.F.: Discrete Space Location–Allocation Solutions from Genetic Algorithms. Annals of Operations Research 6, 35–46 (1986)
15. Kariv, O., Hakimi, S.L.: The p-median problems, in An Algorithmic Approach to Network Location Problems. SIAM Journal on Applied Mathematics 37, 539–560 (1979)
16. Kuehn, A.A., Hamburger, M.J.: A heuristic program for locating warehouses. Management Science 9(4), 643–666 (1963)
17. Levanova, T., Loresh, M.A.: Algorithms of ant system and simulated annealing for the p-median problem. Automation and Remote Control 65, 431–438 (2004)
18. Mladenovic´, N., Brimberg, J., Hansen, P., Moreno-Pérez, J.A.: The p-median problem: A survey of metaheuristic approaches. European Journal of Operational Research 179, 927–939 (2007)
19. Resende, M.G.C., e Werneck, R.F.: A hybrid multistart heuristic for the uncapacitated facility location problem. European Journal of Operational Research 174(1), 54–68 (2005)
20. Rolland, E., Schilling, D.A., Current, J.R.: An Efficient Tabu Search Procedure for the p-Median Problem. European Journal of Operational Research 96, 329–342 (1997)
21. Salhi, S.: Defining tabu list size and aspiration criterion within tabu search methods. Computers and Operations Research 29, 67–86 (2002)
22. Teitz, M.B., Bart, P.: Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph. Operations Research 16, 955–961 (1968)
23. Whitaker, R.: A fast algorithm for the greedy interchange for large-scale clustering and median location problems. Infor. 21, 95–108 (1983)

Using Visual Metrics to Selecting ICA Basis for Image Compression: A Comparative Study

Patricia R. Oliveira[1], Henrique Morimitsu[2], and Esteban F. Tuesta[1]

[1] School of Arts, Science and Humanities - University of Sao Paulo - Brazil
[2] Institute of Mathematics and Statistics - University of Sao Paulo - Brazil

Abstract. In order to obtain a good image compression result, it would be appropriate to previously estimate the error between a distorted image and its reference in such a process. Traditionally, the Mean-Squared Error (MSE) has been used as a standard measure for evaluate the effect of dimensionality reduction methods. More recently, other measures for assessing perceptual image quality has been proposed in the literature. In this paper, the main interest relies on a comparative study between the MSE and the Structural Similarity Index (SSIM), which uses structural similarity as a principle for measuring image quality. The basic aiming for such study is the proposal of an ordering and selecting procedure of the transformation basis found by Independent Component Analysis (ICA), which can take one of these measures into account. The principal motivation for this idea is that, in contrast to Principal Component Analysis (PCA), ICA does not have a property that allows a natural ordering for its components (called ICs). For evaluating the efficiency of such approach, a comparative study between PCA and the ICA-based proposal is also carried out for an image dimensionality reduction application. It can been noted that the ICA method, when using hyperbolic tangent function, could provide an efficient method to select the best ICs.

1 Introduction

For many years, the Mean-Squared Error (MSE) has been the most used quantitative metric for measuring the performance of image compression algorithms. However, recent work has criticized such metric, discussing its serious failures to consider some dependecies (for instance, textures and patterns) that arises in image processing applications [9] [6]. Due its poor performance as a visual metric, some alternatives has been developed by researchers in different signal processing fields.

Instead of using the MSE, recent studies have considered a signal communication framework, in which natural images act as transmitters having the human visual perception system as the receiver [9]. Focusing on this approach, image fidelity measures have been proposed by mathematically modeling each functional perceptual component in such perceptual system and, then integrating these intermediate results in a whole system model. In this sense, the Structural

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 80–89, 2010.

Similarity Index (SSIM) has been proposed for measuring the fidelity of images and other type of signals, such as sound and speech [6].

The basic ideia behind the SSIM approach for measuring image fidelity relies on the fact that the human visual perception system is strongly adapted to capture structural information from visual scenes. Therefore, in compression applications, the retention of the original image structure is an important issue that should be evaluated when comparing a distorted (recovered) image with its reference.

In this work, the MSE and the SSIM metrics are evaluated in an image compression application study which compares Principal Component Analysis (PCA) and Independent Component Analysis (ICA) algorithms. Moreover, these two measures are also used to establish an ordering procedure for finding the most important PCA and ICA basis.

This paper is organized as follows. Section 2 discusses some recent work related to metrics for assesing image quality and PCA/ICA image processing applications. Sections 3 and 4 present discussions of the MSE and the SSIM as image fidelity measures, respectively. Sections 5 and 6 consider basic definitions of PCA and ICA, respectively. In section 7, the ICA basis selection procedure, which can use the MSE or the SSIM measure, is proposed. A comparative study involving the discussed metrics and the application of PCA and ICA for image compression tasks is presented in Section 8. Finally, in section 9, conclusions are drawn, with some questions for further studies.

2 Related Work

A new look at signal fidelity measures was considered in a recent work by Wang and Bovik [9], where they discussed the role of the MSE and other alternative metrics (including the SSIM) for primarily processing image signals. To support the criticism of the MSE, this paper also presented some illustrative examples in a comparative study involving this metric and others for assessing perceptual image quality. More specifically, original images were modified by different types of distortion (noise addition, rotation, JPEG compression, etc.) and then both MSE and SSIM values have been calculated for each altered image. The empirical analyses showed that, in contrast to the SSIM, the MSE values of many of the distorted images were nearly identical, even though these images presented very different visual quality. In the other hand, the images which have been distorted with small geometric modifications presented very large MSE values, yet showed an unperceived loss of visual quality in comparison to the original version.

The basic ideas of the SSIM index have been introduced in [7] and more elaborated in [6] and [8]. To demonstrate the concept of structural similarity, in [6], the authors developed and the SSIM index and showed that it provides favorable results in comparison to other methods used for measuring the quality of JPEG and JPEG2000 compressed images.

In the field of image processing applications, many authors have presented comparative studies between PCA and ICA, often with contradictory results.

A review on such analyses was presented in [2], which showed an independent discussion of PCA and ICA for face recognition. A study on the adequacy of basis learned from data using ICA and how it generalizes to different classes was presented in [5]. In such work, they observed the basis learned from four classes of images (natural, faces, fingerprint, and synthetic) and concluded that the ones learned from natural images have the best generalization. The work developed in [1] addresses the IC selection problem using a procedure that relies on the MSE criterion. They considered applying a cost-complexity function to the MSE values provided by ICA and, next, using the cross validation method to select an appropriate number of ICs. The present work differs from this latest in the sense that we are not concerned in determining the number of components to be used in the dimensionality reduction task. Instead, our objective is to present an approach to ordering the ICs and to chosing the most relevant to be applied to reduce the image dimensionality. Moreover, we want to compare the results obtained by such approach to those obtained by PCA, for different reduction levels and the above discussed visual metrics.

3 Mean Squared Error (MSE)

The goal of an image quality measure is to provide a quantitative score that capture the similarity (or, conversely, the error) between an original version (reference image) and a distorted one.

Given two images $\mathbf{x}$ and $\mathbf{y}$, the MSE between them can be mathematically defined by:

$$MSE(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2, \tag{1}$$

where N is the number of pixels in the images and x_i and y_i are the pixel values in the i-th position of the sample vector.

The main advantages in using the MSE is that it is simple, inexpensive to compute and easy to differentiate, which makes it well suited in context of optimization. Due to such attractive features, or merely by convention, the MSE has been largely applied as a standard measure by which competing image processing algorithms have often been compared. However, in spite of its popularity, this measure has a poor performance as a visual metric, as discussed in [9].

4 Structural Similarity Index (SSIM)

Aiming to overcome the MSE drawbacks, alternative metrics have been recently developed in the signal processing field. One of these approaches that has been proved to be effective is the SSIM index, which takes into account the retention of the original signal structure when evaluating a distorted version.

Mathematically, the SSIM can be defined as follows:

$$SSIM(\mathbf{x},\mathbf{y}) = l(\mathbf{x},\mathbf{y}) \cdot c(\mathbf{x},\mathbf{y}) \cdot s(\mathbf{x},\mathbf{y}) = \frac{2\mu_x\mu_y}{\mu_x^2+\mu_y^2} \cdot \frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2} \cdot \frac{2\sigma_{xy}}{\sigma_x\sigma_y}, \quad (2)$$

where μ_x and μ_y are the local sample means of $\mathbf{x}$ and $\mathbf{y}$, respectively, σ_x and σ_y are the local sample standard deviations of $\mathbf{x}$ and $\mathbf{y}$, respectively, and σ_{xy} is the sample cross correlation of $\mathbf{x}$ and $\mathbf{y}$, after removing their means [9].

In this paper, the local statistics were calculated within a local neigbourhood defined by an 11×11 circular symmetric Gaussian weighting function $\mathbf{w} = \{w_i \mid i = 1, 2 \ldots, N\}$, with standard deviation of 1.5 samples, normalized to unit sum, as defined in [6]. Using such approach, the local statistics can be formulated as:

$$\mu_x = \sum_{i=1}^{N} w_i x_i. \quad (3)$$

$$\sigma_x = \left(\sum_{i=1}^{N} w_i (x_i - \mu_x)^2 \right)^{1/2}. \quad (4)$$

$$\sigma_{xy} = \sum_{i=1}^{N} w_i (x_i - \mu_x)(y_i - \mu_y). \quad (5)$$

As it can be seen in [9] and [11], the SSIM has been proved to perform remarkably well as a visual metric in a variety of image and distortion types. Generally speaking, these studies pointed out that the SSIM measure is more consistent than the MSE measure when considering issues related to human visual perception.

5 Principal Component Analysis (PCA)

One of the most commonly used linear technique for dimensionality reduction is the PCA, which transforms the data in such a way that the variance in the lower-dimensional representation is maximized based on second-order statistics [3].

Essentially, the classical PCA aims to solve an eigenvalue problem:

$$C_x a_j = \lambda_j a_j, \ for\ j = 1, 2, ..., p \quad (6)$$

where C_x is the original data covariance matrix, λ_j is an eigenvalue of C_x and a_j is the eigenvector corresponding to the eigenvalue λ_j. Next, the calculated eigenvalues can be decreasingly ordered: $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$.

The principal components (namely, PCs) can, then, be computed according to the equation:

$$Z_j = a_j^T X = X^T a_j, \ for\ j = 1, 2, ..., p \quad (7)$$

where Z_j is the *j-th* principal component and X represents the original data set. An important property of PCA is that the variances of principal components are the eigenvalues of matrix C. Therefore, dimensionality reduction can be obtained by performing PCA and by keeping only the components with highest variance.

6 Independent Component Analysis (ICA)

The ICA method uses higher-order statistics to project the data into a new subspace whose vectors are as statistically independent as possible [2]. These both techniques have been succesfully applied in image processing, psychometric measurements, among other applications [4].

The noise-free ICA model can be defined as the following linear model:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{8}$$

where the n observed random variables $x_1, x_2, ..., x_n$ are modeled as linear combinations of n random variables $s_1, s_2, ..., s_n$, which are assumed to be statistically mutually independent.

In this sense, ICA model is a generative model, that is, it describes how the observed data are generated by a process of mixing the components s_j. In such model, the independent components s_j cannot be directly observed and the mixing coefficients a_{ij} are also assumed to be unknown. Only the random variables x_i are observed and both the components s_j and the coefficients a_{ij} must be estimated using $\mathbf{x}$.

For simplicity, it is assumed that the unknown mixture matrix $\mathbf{A}$ is square. It simplifies the estimation process, since after estimating matrix $\mathbf{A}$, it is possible to compute its inverse $\mathbf{W}$, and obtain the independent components simply by:

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{9}$$

There are several methods for adapting the mixing mixtures in the ICA model. In this work we used the FastICA algorithm for negentropy maximization, proposed by [4].

7 The Selecting ICA Basis Approach

In contrast to PCA, there is not an ordering scheme for the independent components found by ICA [4]. Even though in some applications this may not be a problem, when performing dimensionality reduction, it is important to establish in advance which components are the most relevant, otherwise there is no guarantee that the loss of information will be minimum for a fixed compression level.

It is known that PCA is optimal in the MSE sense among all linear transformations. Therefore, such measure was firstly chosen for guiding the selection procedure for the ICs. For finding the most important IC (namely, first IC), one can simply use each of the n calculated ICs for compressing the original image and, after calculate the MSE between the recovered and the original image versions. The results of this stage are then used to establish the relevance for all the computed basis.

Considering the previous discussion involving chosing the MSE or the SSIM as a visual metric, a variant of this selection procedure was also developed in

which the ICs are now ordered based on the decreasing SSIM scores obtained for the recovered images, in comparison to the reference (original) image.

In the next section, it can be noted that this simple proposed procedure leads to compression results that are significantly better than those find by PCA, considering the both visual metrics.

8 Experimental Results

In this section one presents the main results obtained in a comparative study involving the MSE and the SSIM metrics for evaluating the performance of PCA and ICA methods for compressing an image set consisting of 23 different classes of images available at Microsoft Research site[1]. For these experiments, one image per class has been randomly chosen.

The preprocessing task involved the arbitrary choice of the number of components and the division of the image in patches corresponding to the selected value. For the experiments of this paper, the number of components to be estimated by both PCA and ICA has been set to 10. Then, each image was divided in 30720 blocks of 10 pixels, producing a matrix with 30720 rows and 10 columns. This matrix was normalized in such a way that each of its cells had its value decreased by the average of the matrix values.

To obtain the principal components, we have used the statistical software Minitab 15.0. The ICA basis have been estimated by the FastICA algorithm and Negentropy proposed by [4], with deflationary ortogonalization. The ICA performance has been tested using two non-linear functions for negentropy: gaussian function and hyperbolic tangent. In this work, we show only the results for hyperbolic tangent, since the use of gaussian function do not lead to convergence for some images.

In order to ascertain whether there are significant differences between the performance of the MSE and the SSIM measures for PCA and ICA results, a test of hypotheses has been conducted. In this sense, the assumptions for applying the test have firstly been checked, which involves verifying the non-normality of the differences between the MSE and the SSIM scores obtained by PCA and ICA, and the dependence condition between the samples of this set. For this study, it was assumed that for each image there were two dependent observations, constituted by the MSE (or the SSIM) values of the principal and independent components.

The non-normality assumption has been tested by using the Anderson-Darling method [10]. This result leads us to the application of non-parametric methods, and in this case, the method selected was the Wilkoxon signed rank test of the median [10]. Table 1 and 2 summarizes this analysis. The results displayed in Table 1 are related to MSE measure (difference between MSE scores for PCA and ICA), and show that the null hypotheses that "the median of the difference between the MSE scores for PCA and ICA is equal to zero" is rejected when

[1] http://research.microsoft.com/

Table 1. MSE used for comparing PCA and ICA(TH)

Diff. of Components	N for Test	Wilcoxon Statistic	p	Estimated Median	Confidence Interval
(1PC-1IC)	23	270	0,0001	1.0180	(0.4300; 2.6000)
(2PC-2IC)	23	156	0.5950	1.0500	(-0.9000;1.0500)
(3PC-3IC)	23	80	0.0800	-0.7319	(-2.6200;0.0700)

Table 2. SSIM used for compare PCA and ICA(TH)

Diff. of Components	N for Test	Wilcoxon Statistic	p	Estimated Median	Confidence Interval
(1IC-1PC)	23	256	0,0001	0.043350	(0.0194; 0.0724)
(2IC-2PC)	23	231	0.0050	0.039760	(0.0156; 0.0619)
(3IC-3PC)	23	156	0.5950	0.002545	(-0.0078;0.0144)

(a) MSE=0, SSIM=1 (b) MSE=377,SSIM=0.4796 (c) MSE=367,SSIM=0.4989

Fig. 1. The example original image (left) and its recovered versions for the first PC (center) and the first IC (right)

considering one and two components, but it is not valid for the case of three components. That indicates that ICA presents better results for image compression rates equal to 80 or 90 percent.

In the other hand, Table 2 is related to SSIM measures (difference between SSIM scores for PCA and ICA), and show that the hypotheses, "the median of the difference between the SSIM scores for ICA and PCA is equal to zero" is rejected when considering one and two components, but it is not valid for the case of three components. Once again, this indicates that ICA presents better results for image compression rates equal to 80 or 90 percent.

In this work, only the first three components have been considered since the cumulated variance proportion for this number of components is just near to 90%.

As an illustrative example, Figures 1, 2 and 3 show one of the original images in the set, together with its related recovered versions for applying PCA and ICA using 1, 2 and 3 components. The MSE and the SSIM values for these recovered images are shown bellow each one.

(a) MSE=173,SSIM=0.7020

(b) MSE=163,SSIM=0.7345

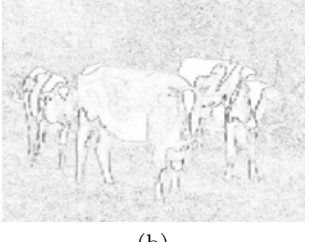
Fig. 2. The recovered image versions for the first two PCs (left) and the first two ICs (right)

(a) MSE=103,SSIM=0.8284

(b) MSE=103,SSIM=0.8316

Fig. 3. The recovered image versions for the first three PCs (left) and the first three ICs (right)

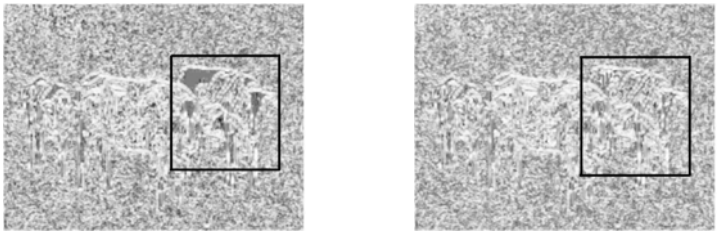
(a)



(b)

Fig. 4. Absolute Error of the recovered image by PCA (a) and ICA (b)

The effectiveness of the SSIM as a visual perceptual metric can be observed by comparing the absolute error maps (see Figure 4) and the SSIM maps (see Figure 5). In both type of maps, brightness corresponds to good values of image fidelity, that is, the brighter the map, the better the quality [9]. This behavior has been perceived for all the images in the set. Therefore, one can afirm that, the SSIM has better captured the similarity between the reference and the recovered image versions.



(a) (b)

Fig. 5. SSIM index map for recovered images by PCA (a) and ICA (b) (with mark space for visibility)

9 Conclusions

In this paper, a selection procedure for ICA basis was proposed, considering two visual metrics. The first measure was the MSE, which has been used as a standard for comparing algorithms for image compression. As this metric suffers from several limitations, another recently developed metric, named SSIM, has also been applied in this selection approach.

A comparative study involving applying the proposed selection techniques evaluated the performance of the MSE and the SSIM as visual metrics for assessing the quality of images compressed by ICA and PCA algorithms. Such analysis showed that the SSIM was able to capture structural similarities between the reference and the recovered image versions, presenting also better values of image fidelity than that obtained by the MSE metric.

Moreover, the empirical analysis showed that the ICA compression method can be used as an alternative technique to PCA, since ICA provided favorable results in comparison to those obtained by PCA.

As future work, some experiments are being conducted in order to evalute the performance of different visual perceptual metrics for another image processing techniques, such as image segmentation and classification.

References

1. Wu, E.H., Yu, P.L.H., Li, W.K.: An Independent Component Ordering and Selection Procedure Based on the MSE Criterion. In: Rosca, J.P., Erdogmus, D., Príncipe, J.C., Haykin, S. (eds.) ICA 2006. LNCS, vol. 3889, pp. 286–294. Springer, Heidelberg (2006)
2. Delac, K., Grgic, M., Grgic, S.: Independent Comparative Study of PCA, ICA, and LDA on the Feret Data Set. International Journal of Imaging Systems and Technology 15(5), 252–260 (2005)
3. Jolliffe, I.I.: Principal Component Analysis, 2nd edn., USA. Springer Series in Statistics (2002)
4. Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis, p. 421. Wiley-Interscience, Hoboken (2001)

5. Ferreira, A.J., Figueiredo, M.A.T.: On the use of independent component analysis for image compression. Signal Processing: Image Communication 21, 378–389 (2006)
6. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)
7. Wang, Z., Bovik, A.C.: A Universal image quality index. IEEE Signal Processing Letters 9(3), 81–84 (2002)
8. Wang, Z., Bovik, A.C.: Modern Image Quality Assessment. Morgan & Claypool, San Rafael (2006)
9. Wang, Z., Bovik, A.C.: Mean Square Error: love or leave it? A new look at signal fidelity measures. IEEE Signal Processing Magazine 98 (January 2009)
10. Hollander, M., Wolfe, D.A.: Nonparametric Statistical Methods, 2nd edn. John Wiley & sons, Inc., Chichester (1999)
11. Sheikh, H.R., Sabir, M.F., Bovik, A.C.: A statistical evaluation of recent full reference image quality assessment algorithms. IEEE Transactions on Image Processing, 15(11), 3449–3451 (2006)

Extending DeLP with Attack and Support for Defeasible Rules*

Andrea Cohen, Alejandro J. Garcia, and Guillermo R. Simari

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Artificial Intelligence Research and Development Laboratory,
Department of Computer Science and Engineering - Universidad Nacional del Sur,
Bahía Blanca, Argentina
{ac,ajg,grs}@cs.uns.edu.ar

Abstract. In several formalisms of classical argumentation two kinds of defeaters are considered: rebutting and undercutting. The former represents an attack to a conclusion and the latter an attack to an inference step. Defeasible Logic Programming (DeLP) is a formalism that combines argumentation and logic programming and provides a concrete argumentation system where arguments are built using program rules and facts. DeLP allows to identify arguments whose conclusions or intermediate conclusions are in contradiction. In that way, rebutting defeaters are captured. Nevertheless, in DeLP is not possible to represent an explicit attack to a program rule in order to capture undercutting defeaters. The contribution of this work is to extend the formalism of DeLP in order to allow the representation of both support and attack for defeasible rules.Therefore, it will be possible to build arguments that provide reasons for or against a defeasible rule and thus, undercutting defeaters and backings could be constructed.

1 Introduction

Argumentation is a form of reasoning where a claim is accepted or rejected according to the analysis of the arguments for and against it. When a rule supporting a conclusion may be defeated by new information it is said that such reasoning is *defeasible* [3,6]. As explained in [1], when defeasible reasons are chained to reach a conclusion, *arguments* instead of proofs are obtained. Pollock [6,7] postulates that reasoning operates in terms of *reasons* that can be assembled to comprise *arguments*. He distinguishes two kinds of reasons, *nondefeasible* and *defeasible*, and establishes that defeasible reasons have defeaters: rebutting and undercutting defeaters. *Rebutting defeaters* attack the conclusion of an inference by supporting the opposite one (*i. e.*, they are reasons for denying the conclusion). *Undercutting defeaters* attack the connection between the premises and the conclusion without attacking the conclusion directly. For a conclusion to be accepted is not enough to have an argument for it. A claim is accepted if it emerges undefeated from a justificatory process.

In his book *The Uses of Argument* [8], Toulmin proposed a model for the layout of arguments. He argued that arguments need to be analyzed using a richer format than the traditional one of formal logic. Whereas a formal logic analysis uses the dichotomy of

* Partially supported by UNS (PGI 24/ZN18) and CONICET (PIP 112-200801-02798).

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 90–99, 2010.

premises and conclusions, Toulmin proposed a scheme that in addition to data and claim distinguishes four elements: warrant, backing, rebuttal and qualifier (see Figure 1). The *claim* is the original assertion that we are committed to and must justify when challenged. The *data* provides the basis of the claim. The *warrant* is an inference license that provides the connection between data and claim. The *backing* provides support for the warrant and it shows why the warrant holds. The *rebuttal* provides conditions of exception for the argument. The last element is the *qualifier*, which indicates the strength of the step from data to claim, as conferred by the warrant.

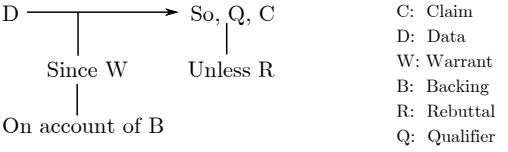


Fig. 1. The general form of Toulmin's scheme for the layout of arguments

Defeasible Logic Programming (DeLP) [2] is a formalism that combines argumentation and logic programming. It provides a concrete argumentation system that is fully implemented and available online [5]. DeLP allows to identify arguments whose conclusions or intermediate conclusions are in contradiction, capturing the notion of rebutting defeater. Nevertheless, in DeLP is not possible to represent explicit attack and explicit support for defeasible rules in order to capture Pollock's undercutting defeaters and Toulmin's backings. The contribution of this work is to extend the formalism of DeLP to capture these notions, allowing to build arguments that provide reasons for or against a defeasible rule, which increases the expressive power of DeLP.

The rest of this paper is organized as follows. In Section 2 we present an overview of DeLP and we introduce the motivation of this work with an illustrating example. In Section 3 we present an extension to the representational language of DeLP and in Section 4 we introduce the concepts of defeasible derivation, argument and undercutting defeater. Finally, in Section 5 conclusions and related works are commented.

2 Background and Motivation

A brief explanation of DeLP is included below (see [2] for full details). As in Logic Programming, in DeLP knowledge is represented using facts and rules. In addition, it has the declarative capability of representing weak information in the form of *defeasible rules*, and a defeasible argumentation inference mechanism for warranting the entailed conclusions. A defeasible logic program is a set of facts, strict rules and defeasible rules, defined as follows. *Facts* are ground literals representing atomic information, or the negation of atomic information using the strong negation "$\sim$" (*e. g.* $\sim electricity$ or day). *Strict Rules* represent non-defeasible information and are denoted $L_0 \leftarrow L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals (*e. g.* $\sim night \leftarrow day$ or $dark \leftarrow \sim illuminated$). *Defeasible Rules* represent tentative information that may be used if nothing could be posed against it, and are denoted $L_0 \prec L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground

literals (*e. g.* $lights\_on \prec switch\_on$ or $\sim lights\_on \prec \sim electricity$). A defeasible rule "$Head \prec Body$" expresses that *"reasons to believe in the antecedent Body give reasons to believe in the consequent Head"*. When required, a program $\mathcal{P}$ is denoted as (Π, Δ) distinguishing the subset Π of facts and strict rules, and the subset Δ of defeasible rules. From a program $\mathcal{P}$ contradictory literals could be derived, since *strong negation* is allowed in the head of rules.

For the treatment of contradictory knowledge DeLP incorporates a defeasible argumentation formalism. This formalism allows the identification of the pieces of knowledge that are in contradiction, and a *dialectical process* is used for deciding which information prevails as warranted. The dialectical process involves the construction and evaluation of arguments that either support or interfere with the query under analysis. Briefly, an *argument* for a literal h, denoted $\langle \mathcal{A}, h \rangle$ is a minimal set $\mathcal{A}$ of defeasible rules, such that $\mathcal{A} \cup \Pi$ is non-contradictory and there is a derivation for h from $\mathcal{A} \cup \Pi$. A literal h is *warranted* if there exists a non-defeated argument $\mathcal{A}$ supporting h. To establish if $\langle \mathcal{A}, h \rangle$ is a non-defeated argument, *defeaters* for $\langle \mathcal{A}, h \rangle$ are considered. A detailed explanation of the dialectical process of DeLP is not included here due to space restrictions, but it can be found in [2].

The following example, given by Pollock in [7], shows clearly that DeLP can not represent undercutting defeaters properly. Pollock's example states that if an object looks red ("$looks\_red$"), then there is a defeasible reason for believing that the object is red ("is_red"). Nevertheless, if we know that illumination by red lights can make an object look red when it is not, and we see an object that looks red and it is illuminated by red lights, we should refrain from concluding that it is red though it might be red. Thus, that the object is illuminated by red lights ("$illuminated\_by\_red\_lights$") is an undercutting defeater for this inference, since it is not a reason for denying that the object is red, but it is a reason for denying that the object would not look red unless it were red.

For representing Pollock's example in DeLP consider the following set of rules: $\Delta_a = \{(is\_red \prec looks\_red), (\sim is\_red \prec looks\_red, illuminated\_by\_red\_lights)\}$.[1] If the set of facts is $\Pi_a = \{looks\_red\}$, then from (Π_a, Δ_a) the literal "is_red" is warranted. Suppose now that we have additional information indicating that the object is illuminated by red lights. This new situation can be represented by the set of facts $\Pi_b = \{looks\_red, illuminated\_by\_red\_lights\}$. In this case, from (Π_b, Δ_a) the literal "$\sim is\_red$" is warranted. As in Pollock's example, from (Π_b, Δ_a) the literal "is_red" is no longer warranted, however, in contrast with Pollock's example, now a new literal is warranted: "$\sim is\_red$". The problem is that in DeLP undercutting defeaters can not be represented. Note that the rule "$\sim is\_red \prec looks\_red, illuminated\_by\_red\_lights$" expresses that if the object looks red and it is illuminated by red lights, there are reasons to believe that the object is not red. By this rule we conclude that the object is not red, leading to a rebutting defeater instead of an undercutting defeater. If we want to correctly represent this situation in DeLP we should be able to express reasons against the defeasible rule "$is\_red \prec looks\_red$". Nevertheless, in DeLP it is not possible to represent attack for defeasible rules, since the head of a rule must be a single literal. Note also that in DeLP it is not possible to represent support for defeasible rules.

[1] When required, parenthesis will be used for distinguishing one rule from another.

In this work, we will extend the formalism of DeLP in order to represent attack and support for defeasible rules. Thus, Pollock's undercutting defeaters and Toulmin's backings can be expressed.

3 Extended Language for Supporting and Attacking Rules

In this section we will introduce the syntax of Extended Defeasible Logic Programming (E-DeLP), which is an extension of the language of DeLP defined in [2]. The E-DeLP language is defined in terms of four disjoint sets: a set of *facts*, a set of *strict rules*, a set of *defeasible rules* and a set of *defeasible meta-rules*. A literal "L" is considered to be a *ground atom* "A" or a *negated ground atom* "$\sim A$" (where "$\sim$" represents the *strong negation*), and can be used as a fact or as an element of a rule.

In E-DeLP facts and strict rules represent non-defeasible or indisputable information, whereas defeasible rules and meta-rules are used to represent defeasible knowledge (*i. e.*, tentative information that may be used if nothing could be posed against it). Another difference is that defeasible rules and defeasible meta-rules have a label. This label will be used for identifying the rules and thus, they provide a way for supporting and attacking them. Therefore, labels can not be used arbitrarily in rules. The main difference between defeasible rules and defeasible meta-rules is that a defeasible meta-rules can have a label or a negated label in their head (representing support and attack for a rule respectively), whereas defeasible rules can not.

The representation of facts in E-DeLP is the same as in DeLP. Strict rules and defeasible rules are different from their counterparts in DeLP, and defeasible meta-rules are the key element of our proposed extension. For the representation of defeasible rules and defeasible meta-rules we consider a distinguished set of atoms, called $\mathcal{L}abels$, that will be used for denoting rule's labels. If an atom "R" is an element of $\mathcal{L}abels$ there must exist a defeasible rule or a defeasible meta-rule labeled "R". Given a literal "L", $atom(L)$ represents the atom associated to "L" (*i. e.* $atom(\sim a) = a$ and $atom(a) = a$).

Definition 1 (Strict Rule). *A Strict Rule is an ordered pair, denoted "$Head \leftarrow Body$", where $Head$ is a literal that verifies $atom(Head) \notin \mathcal{L}abels$ and $Body$ is a finite non-empty set of literals such that for all $L_i \in Body$ it holds that $atom(L_i) \notin \mathcal{L}abels$. A strict rule with head L_0 and body $\{L_1, \ldots, L_n\}$ can also be written as $L_0 \leftarrow L_1, \ldots, L_n$ $(n \geq 1)$.*

Definition 2 (Defeasible Rule). *A Defeasible Rule is an ordered triplet, denoted "$Head$ -$(Label)\prec$ $Body$", where $Head$ is a literal that verifies $atom(Head) \notin \mathcal{L}abels$, $Body$ is a finite set of literals such that for all $L_i \in Body$ it holds that $atom(L_i) \notin \mathcal{L}abels$, and $Label \in \mathcal{L}abels$. A defeasible rule with head L_0, body $\{L_1, \ldots, L_n\}$ and label R can also be written as L_0 -$(R)\prec$ $L_1, \ldots, L_n$ $(n \geq 1)$. A defeasible rule with head L_0, empty body and label R can also be written as L_0 -$(R)\prec$.*

Definition 3 (Defeasible Meta-Rule). *A Defeasible Meta-Rule is a rule that verifies one of the following conditions:*

(a) Is an ordered triplet, denoted "$Head$ -$(Label)\prec$ $Body$", where $Head$ is a literal that verifies $atom(Head) \in \mathcal{L}abels$, $Body$ is a finite non-empty set of literals such that

for all $L_i \in Body$ it holds that $atom(L_i) \notin \mathcal{L}abels$, and $Label \in \mathcal{L}abels$ verifies $Label \neq atom(Head)$. A defeasible meta-rule with head L_0, body $\{L_1, \ldots, L_n\}$ and label R can also be written as $L_0 \text{ -}(R)\prec L_1, \ldots, L_n$ $(n \geq 1)$.

(b) Is a defeasible rule with an empty body and no label, denoted "$Head \prec$ ", where $Head$ is a literal that verifies $atom(Head) \in \mathcal{L}abels$.

A defeasible meta-rule without an empty body must satisfy $Label \neq atom(Head)$. The aim of this constraint is to avoid the construction of defeasible meta-rules like "$r \text{ -}(r)\prec b$" or "$\sim r \text{ -}(r)\prec b$". A defeasible meta-rule "$label \prec$ " would express that "*there are (defeasible) reasons to believe in* $label$", and it is called a presumption.

Definition 4 (Extended Defeasible Logic Program). *An Extended Defeasible Logic Program (e-de.l.p.) $\mathcal{P}$ is a possibly infinite set of facts, strict rules, defeasible rules and meta-rules. When required, we will denote $\mathcal{P}$ as (Π, Δ, Σ) distinguishing the subset Π of facts and strict rules, the subset Δ of defeasible rules, and the subset Σ of defeasible meta-rules.*

A program $\mathcal{P}$=(Π, Δ, Σ) is a valid *e-de.l.p.* if for every pair of rules $R_i, R_j \in (\Delta \cup \Sigma)$ labeled E_i and E_j respectively, it holds that $E_i \neq E_j$. This constraint is to provide a unique way for identifying the rules. In E-DeLP, the subset Π of facts and strict rules must be non-contradictory. Recall that a set of rules S is contradictory if there exists a derivation for a pair of complementary literals from S.

The following two examples show how the proposed language of E-DeLP allows the representation of explicit attack or explicit support for defeasible rules. Example 1 shows a possible formulation of Pollock's example introduced above, where an attack to a defeasible rule is expressed and thus, an undercutting defeater can be constructed. Example 2 shows how to represent support for a defeasible rule.

Example 1. *A possible representation for the situation described in the above Pollock's example is given by the e-de.l.p. $\mathcal{P}_1$=$(\Pi_1, \Delta_1, \Sigma_1)$:*

$$\Pi_1 = \left\{ \begin{array}{l} looks\_red \\ illuminated\_by\_red\_lights \end{array} \right\} \quad \Sigma_1 = \left\{ \begin{array}{l} \sim r_1 \text{ -}(r_2)\prec illuminated\_by\_red\_lights \\ r_1 \prec \\ r_2 \prec \end{array} \right\}$$

$$\Delta_1 = \left\{ is\_red \text{ -}(r_1)\prec looks\_red \right\}$$

where Π_1 represents that there is an object that looks red and it is illuminated by red lights. The defeasible rule "$is\_red \text{ -}(r_1)\prec looks\_red$" expresses that the object looking red provides reasons to believe that it is red. The defeasible meta-rule "$\sim r_1 \text{ -}(r_2)\prec illuminated\_by\_red\_lights$" provides a reason against the defeasible rule "$is\_red \text{ -}(r_1)\prec looks\_red$", and expresses that if the object is illuminated by red lights there are reasons to believe that the rule r_1 should not be taken into account to decide whether the object is red. Observe that the presumptions "$r_1 \prec$ " and "$r_2 \prec$ " provide support for the rules r_1 and r_2 respectively.

Next, we introduce Toulmin's famous example about Harry in order to illustrate support for defeasible rules. In that example, Toulmin discusses the claim that Harry is a British

subject ("$british\_subject$"). The claim is supported by the data that Harry was born in Bermuda ("$born\_in\_bermuda$"). The connection between data and claim is expressed by the warrant that a man born in Bermuda will generally be a British subject. The warrant can be supported by the backing that there are certain statutes and other legal provisions to that effect ("$legal\_support$"). Given that the warrant does not have total justifying force, the claim that Harry is a British subject must be qualified: it follows presumably. A possible rebuttal for the claim is that both Harry's parents were aliens ("$alien\_parents$").

Example 2. *A possible representation in E-DeLP for Toulmin's example about Harry is given by the e-de.l.p. $\mathcal{P}_2$=($\Pi_2, \Delta_2, \Sigma_2$):*

$$\Pi_2 = \{born\_in\_bermuda, legal\_support, alien\_parents\}$$

$$\Delta_2 = \left\{ \begin{array}{l} british\_subject \text{ -}(r_3)\prec born\_in\_bermuda \\ \sim british\_subject \text{ -}(r_4)\prec alien\_parents \end{array} \right\} \quad \Sigma_2 = \left\{ \begin{array}{l} r_3 \text{ -}(r_5)\prec legal\_support \\ r_4 \prec \\ r_5 \prec \end{array} \right\}$$

Here, the data "$born\_in\_bermuda$", the backing "$legal\_support$" and the rebuttal "$alien\_parents$" are facts, since they represent information that is effectively known. The warrant is given by the defeasible rule "$british\_subject$ -$(r_3)\prec$ $born\_in\_bermuda$", since reasons to believe that Harry was born in Bermuda provide reasons to believe that he is a British subject. Note that in this representation there is no qualifier for the claim "$british\_subject$" since Toulmin's qualifier "probably" is considered to be implicit in the defeasibility of the rule r_3. The rule "r_3 -$(r_5)\prec$ $legal\_support$" represents that the backing provides support for the warrant (i. e., support for the defeasible rule r_3). Finally, the rebuttal "$alien\_parents$" that attacks the claim is represented by the defeasible rule "$\sim british\_subject$ -$(r_4)\prec$ $alien\_parents$".

Observe that defeasible meta-rules provide support or attack for defeasible rules, or for another defeasible meta-rules. For simplicity, in the examples we only use defeasible meta-rules that provide support or attack for defeasible rules.

4 Arguments and Undercutting Defeaters in E-DeLP

In this section we will introduce the concepts of defeasible derivation, argument and undercutting defeater. First, we introduce the concept of defeasible derivation. Later, we present the notion of argument, which is a minimal and non-contradictory set of rules used to derive a conclusion. Finally, a categorization of defeaters is provided, introducing the notion of undercutting defeater.

Definition 5 (Defeasible Derivation). *Let $\mathcal{P}$=(Π, Δ, Σ) be an e-de.l.p. and L a literal. A defeasible derivation of L from $\mathcal{P}$, denoted $\mathcal{P} \mid\sim L$, consists of a finite sequence $L_1, L_2, \ldots, L_n = L$ of literals, where each literal L_i is in the sequence because:*

(a) L_i is a fact in Π, or there exists a defeasible meta-rule in Σ with head L_i, empty body and no label,
(b) there exists a strict rule in Π with head L_i and body $B_1, B_2, \ldots, B_k$ ($k \geq 1$), and every B_t ($1 \leq t \leq k$) is an element L_j of the sequence appearing before L_i ($j < i$),

(c) *there exists a defeasible rule in* Δ *with head* L_i*, body* $B_1, B_2, \ldots, B_k$ $(k \geq 1)$ *and label* R*, where* R *and every* B_t $(1 \leq t \leq k)$ *is an element* L_j *of the sequence appearing before* L_i $(j < i)$*,*

(d) *there exists a defeasible rule in* Δ *with head* L_i*, empty body and label* R*, where* R *is an element* L_j *of the sequence appearing before* L_i $(j < i)$*, or*

(e) *there exists a defeasible meta-rule in* Σ *with head* L_i*, body* $B_1, B_2, \ldots, B_k$ $(k \geq 1)$ *and label* R*, where* R *and every* B_t $(1 \leq t \leq k)$ *is an element* L_j *of the sequence appearing before* L_i $(j < i)$*.*

A defeasible derivation for a literal L is called a *strict derivation* if either L is a fact or every rule used for obtaining the derivation sequence is a strict rule. Conditions (c), (d) and (e) of the above definition establish that if a literal L_i appearing on the derivation sequence is the head of a defeasible rule or a defeasible meta-rule labeled R, then R must appear in the sequence before L_i. This implies that, before using a defeasible rule or a defeasible meta-rule for deriving the literal in its head, the rule must be *applicable* (*i. e.* there must exist a previous defeasible derivation for the rule label). Note that Definition 3 states that defeasible meta-rules with an empty body do not have a label. The reason for this differentiation is the need for certain meta-rules to be initially regarded as applicable.

Example 3. *Let* $\mathcal{P}_3 = (\Pi_3, \Delta_3, \Sigma_3)$ *be the following e-de.l.p.:*

$$\Pi_3 = \left\{ \begin{array}{lll} a & c \leftarrow a \\ b & r_6 \\ \sim e \end{array} \right\} \quad \Delta_3 = \left\{ \begin{array}{l} \sim d \prec_{(r_6)} c \\ d \prec_{(r_7)} b \end{array} \right\} \quad \Sigma_3 = \left\{ \begin{array}{l} r_7 \prec_{(r_8)} \sim e \\ r_8 \prec \end{array} \right\}$$

Some defeasible and strict derivations that can be obtained from $\mathcal{P}_3$ *follows:*

- *The sequence "b,* $\sim e$*,* r_8*,* r_7*, d" is a defeasible derivation for the literal "d", obtained from the following set of rules in* $\mathcal{P}_3$*:* $\{(r_8 \prec), (r_7 \prec_{(r_8)} \sim e), (d \prec_{(r_7)} b)\}$.
- *The sequence "a, c" is a strict derivation for the literal "c" obtained from the following set of rules in* $\mathcal{P}_3$*:* $\{c \leftarrow a\}$.

The definition of argument given in [2] is extended and adapted to fit the formalism of E-DeLP, since additional elements (such as defeasible meta-rules) were introduced.

Definition 6 (Argument Structure). *Let* h *be a literal and* $\mathcal{P} = (\Pi, \Delta, \Sigma)$ *an e-de.l.p.. We say that* $\langle \mathcal{A}, h \rangle$ *is an argument structure for* h *if:*

1. $\mathcal{A} \subseteq \Delta \cup \Sigma$*,*
2. *there exists a defeasible derivation for* h *from* $\Pi \cup \mathcal{A}$*,*
3. *the set* $\Pi \cup \mathcal{A}$ *is non-contradictory, and*
4. $\mathcal{A}$ *is minimal: there is no proper subset* $\mathcal{A}'$ *of* $\mathcal{A}$ *such that* $\mathcal{A}'$ *satisfies conditions (1), (2) and (3).*

For instance, in Example 1 $\langle \{(is\_red \prec_{(r_1)} looks\_red), (r_1 \prec)\}, is\_red \rangle$ is an argument structure for the literal "is_red", obtained from $\mathcal{P}_1$. Then, in Example 2 two argument structures that can be obtained from $\mathcal{P}_2$ are the following: $\langle \mathcal{A}_1, british\_subject \rangle$ is an argument for the literal "$british\_subject$", where $\mathcal{A}_1$ is $\{(british\_subject \prec_{(r_3)} born\_in\_bermuda), (r_3 \prec_{(r_5)} legal\_support), (r_5 \prec)\}$, and $\langle \mathcal{A}_2, \sim british\_subject \rangle$ is an argument for the literal "$\sim british\_subject$", where $\mathcal{A}_2$ is $\{(\sim british\_subject \prec_{(r_4)} alien\_parents), (r_4 \prec)\}$.

Proposition 1. *If an extended defeasible logic program $\mathcal{P}$ has no facts or defeasible meta-rules with an empty body, then no argument structure can be obtained from $\mathcal{P}$.*

Proof. Straightforward from Definitions 5 and 6. □

Example 4. *Consider the following scenario, where a prisoner is being judged for a crime. Generally, if a prisoner plead himself guilty we can assume that he is guilty. Nevertheless, if the prisoner has been threatened before his confession, we can not ensure that his statement was not influenced by the threat. In this case, we should refrain from concluding that the prisoner is guilty based on his plead. However, if the judge plead the prisoner guilty based on the provided evidence, we have new reasons to believe that the prisoner is guilty. A possible representation in E-DeLP for this situation is given by the e-de.l.p. $\mathcal{P}_4$=$(\Pi_4, \Delta_4, \Sigma_4)$:*

$$\Pi_4 = \left\{ \begin{array}{l} prisoner\_confessed\_its\_guilty \\ threatened\_prisoner \\ judge\_plead\_prisoner\_guilty \end{array} \right\} \quad \Sigma_4 = \left\{ \begin{array}{l} \sim r_9 \text{ -}(r_{10})\!\prec threatened\_prisoner \\ r_9 \prec \\ r_{10} \prec \\ r_{11} \prec \end{array} \right\}$$

$$\Delta_4 = \left\{ \begin{array}{l} guilty\_prisoner \text{ -}(r_9)\!\prec prisoner\_confessed\_its\_guilty \\ guilty\_prisoner \text{ -}(r_{11})\!\prec judge\_plead\_prisoner\_guilty \end{array} \right\}$$

The argument structure $\langle\{(\sim r_9 \text{ -}(r_{10})\!\prec threatened\_prisoner), (r_{10} \prec)\}, \sim r_9\rangle$ for the literal "$\sim r_9$" can be obtained from $\mathcal{P}_4$. Two argument structures that can be obtained for the literal "$guilty\_prisoner$" follows:

$$\langle\{(guilty\_prisoner \text{ -}(r_9)\!\prec prisoner\_confessed\_its\_guilty), (r_9 \prec)\}, guilty\_prisoner\rangle$$
$$\langle\{(guilty\_prisoner \text{ -}(r_{11})\!\prec judge\_plead\_prisoner\_guilty), (r_{11} \prec)\}, guilty\_prisoner\rangle$$

Given an *e-de.l.p.* $\mathcal{P}$=(Π, Δ, Σ) we will say that two literals h_1 and h_2 *disagree* if the set $\Pi \cup \{h, h_1\}$ is contradictory. Let $\langle\mathcal{A}_1, h_1\rangle$ and $\langle\mathcal{A}_2, h_2\rangle$ be two argument structures obtained from $\mathcal{P}$, we will say that $\langle\mathcal{A}_1, h_1\rangle$ is a *counter-argument* for $\langle\mathcal{A}_2, h_2\rangle$ if there exists a sub-argument $\langle\mathcal{A}, h\rangle$ of $\langle\mathcal{A}_2, h_2\rangle$ such that h and h_1 disagree. An argument structure $\langle\mathcal{B}, q\rangle$ is a *sub-argument* structure of $\langle\mathcal{A}, h\rangle$ if $\mathcal{B} \subseteq \mathcal{A}$.

An argument $\langle\mathcal{A}_1, h_1\rangle$ is a *defeater* for an argument $\langle\mathcal{A}_2, h_2\rangle$ if $\langle\mathcal{A}_1, h_1\rangle$ is a counter-argument for $\langle\mathcal{A}_2, h_2\rangle$ and $\langle\mathcal{A}_2, h_2\rangle$ is not better than $\langle\mathcal{A}_1, h_1\rangle$ with respect to a given comparison criterion. In particular $\langle\mathcal{A}_1, h_1\rangle$ is an *undercutting defeater* for $\langle\mathcal{A}_2, h_2\rangle$if $h_1 = \sim R$, where $R \in \mathcal{L}abels$, otherwise $\langle\mathcal{A}_1, h_1\rangle$ is a *rebutting defeater* for $\langle\mathcal{A}_2, h_2\rangle$. That is, an argument $\langle\mathcal{A}_1, h_1\rangle$ is an undercutting defeater if it is an argument for a negated label, and it is a rebutting defeater if either $atom(h_1) \notin \mathcal{L}abels$ or h_1 is a non-negated label.

Similarly to [2], the comparison criterion in E-DeLP is modular and thus, the most appropriate criterion for the domain that is being represented can be selected. Since E-DeLP provides support and attack for defeasible rules, the argument comparison criterion used for deciding between two competing arguments should handle this new representation correctly. Several comparison criteria can be used, for instance, *generalized specificity* [2] can be adapted to this formalism or a criterion based on rule's priorities can be defined. However, due to space restrictions we do not include the formal definitions here. The following example illustrate the notions introduced above.

Example 5. *Let us consider again Pollock's example, and suppose that we have additional information indicating that the object has recently been painted white. We know that an object being painted white provides reasons to believe that the object is white. Furthermore, if we know that the object is white, we certainly know that the object is not red. This new situation is represented by the e-de.l.p.* $\mathcal{P}_5$=$(\Pi_5, \Delta_5, \Sigma_5)$*, which is an extended version of the program* $\mathcal{P}_1$ *from Example 1:*

$$\Pi_5 = \{looks\_red, illuminated\_by\_red\_lights, painted\_white, \sim is\_red \leftarrow is\_white\}$$
$$\Delta_5 = \{(is\_red \text{ -}(r_{12})\prec looks\_red), (is\_white \text{ -}(r_{14})\prec painted\_white)\}$$
$$\Sigma_5 = \{(\sim r_{12} \text{ -}(r_{13})\prec illuminated\_by\_red\_lights), (r_{12} \prec), (r_{13} \prec), (r_{14} \prec)\}$$

The argument structure $\langle \mathcal{A}_3, \sim is\_red\rangle$ *is a counter-argument for* $\langle \mathcal{A}_4, is\_red\rangle$ *and vice-versa, where* $\mathcal{A}_3 = \{(is\_white \text{ -}(r_{14})\prec painted\_white), (r_{14} \prec)\}$ *and* $\mathcal{A}_4 = \{(is\_red \text{ -}(r_{12})\prec looks\_red), (r_{12} \prec)\}$.

The argument structure $\langle \mathcal{A}_5, \sim r_{12}\rangle$ *is a counter-argument for* $\langle \mathcal{A}_4, is\_red\rangle$ *since* $\langle \mathcal{A}_6, r_{12}\rangle$ *is a sub-argument of* $\langle \mathcal{A}_4, is\_red\rangle$*, and the literals "*$r_{12}$*" and "*$\sim r_{12}$*" disagree, where* $\mathcal{A}_5$ =$\{(\sim r_{12} \text{ -}(r_{13})\prec illuminated\_by\_red\_lights), (r_{13} \prec)\}$ *and* $\mathcal{A}_6$ =$\{r_{12} \prec \}$.

Assuming that the comparison criterion is such that $\mathcal{A}_4$ *is better than* $\mathcal{A}_3$*, and* $\mathcal{A}_5$ *is better than* $\mathcal{A}_6$*, we have that* $\langle \mathcal{A}_5, \sim r_{12}\rangle$ *is an undercutting defeater for* $\langle \mathcal{A}_4, is\_red\rangle$*, and* $\langle \mathcal{A}_4, is\_red\rangle$ *is a rebutting defeater for* $\langle \mathcal{A}_3, \sim is\_red\rangle$.

Recall that in DeLP a dialectical process is carried out for deciding which information prevails as warranted. That process involves the construction and evaluation of arguments that either support or interfere with the query under analysis. Given that we have adapted the notions of argument structure and defeater, in E-DeLP we will use the same dialectical process as in DeLP, without requiring any modifications. For instance, in Example 5, $\langle \mathcal{A}_3, \sim is\_red\rangle$ is defeated by $\langle \mathcal{A}_4, is\_red\rangle$, and $\langle \mathcal{A}_4, is\_red\rangle$ is in turn defeated by $\langle \mathcal{A}_5, \sim r_{12}\rangle$. Then, $\langle \mathcal{A}_3, \sim is\_red\rangle$ is reinstated and thus, the literal "$\sim is\_red$" is warranted from $\mathcal{P}_5$. The following Proposition shows that E-DeLP is an extension of DeLP.

Proposition 2. *Given a defeasible logic program* $\mathcal{P}_1$*, there exists an e-de.l.p.* $\mathcal{P}_2$ *such that if a literal* L *is warranted from* $\mathcal{P}_1$*, then* L *is warranted from* $\mathcal{P}_2$.

Proof sketch. The set of facts and strict rules in $\mathcal{P}_2$ is the same as in $\mathcal{P}_1$. We replace every defeasible rule r_i = "$h_i \prec b_i$" in $\mathcal{P}_1$ for a pair of rules in $\mathcal{P}_2$: a defeasible rule "h_i -$(l_i)\prec$ b_i" and a defeasible meta-rule "$l_i \prec$ " □

In addition to the representation of undercutting defeaters and backings, in E-DeLP it is possible to represent everything that was represented in DeLP. The above Proposition shows that E-DeLP effectively increases the expressive power of DeLP.

5 Conclusions and Related Work

In this paper an extension of DeLP has been proposed. This extension introduces the representation of defeasible meta-rules that allows both support and attack for defeasible rules or meta-rules. Thus, undercutting defeaters and backings can be represented.

Therefore, E-DeLP has more expressive power than DeLP and can capture two notions that are absent in the formalism described in [2].

In [11] Verheij performed a reconstruction of Toulmin's ideas. The starting point for Verheij's reconstruction is a theory called DEFLOG [9,10]. The theory is sentence-based and focuses on the defeat of prima facie justified assumptions, instead on the defeat of the arguments in terms of them. Briefly, its logical language has two connectives $\times$ (dialectical negation) and $\sim>$ (primitive implication). One difference between Verheij's approach and our extension is that DEFLOG is sentence-based whereas E-DeLP is based on logic programming. In addition, arguments in DEFLOG are sets of statements, whereas in E-DeLP arguments are sets of specific rules. In DEFLOG it is possible to combine and nest the connectives $\times$ and $\sim>$ in order to obtain more complex statements and thus, allowing to represent both Toulmin's backings and Pollock's undercutting defeaters. Nevertheless, since dialectical negation expresses defeat, a fixed criterion for argument comparison is used: an argument for a statement $\times S$ will always be preferred to an argument for a statement S. Thus, in Verheij's approach it is not possible to express attack without defeat. On the contrary, in E-DeLP it is possible to represent both attack and defeat, the existence of a counter-argument does not necessarily imply defeat.

Nute's d-Prolog [3,4] was the first to introduce defeasible reasoning programming with strict and defeasible rules. That language also provides *defeater rules* like *"sick birds do not fly"*, that can not be used for entail conclusions but to account for the exceptions to defeasible rules. Although defeater rules can be used for simulating in some sense undercutting defeaters, they are not stating a reasoning against a rule: they provide a reason against a literal. One of the main differences with our approach is that in d-Prolog there is no notion of argument and therefore, no notion of defeater. In order to decide between contradictory conclusions in d-Prolog, pairs of rules are compared.

References

1. Chesñevar, C.I., Maguitman, A.G., Loui, R.P.: Logical models of argument. ACM Computing Surveys 32(4), 337–383 (2000)
2. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming 4(1-2), 95–138 (2004)
3. Nute, D.: Defeasible reasoning: a philosophical analysis in PROLOG. In: Fetzer, J.H. (ed.) Aspects of Artificial Intelligence, pp. 251–288. Kluwer Academic Pub., Dordrecht (1988)
4. Nute, D.: Defeasible logic. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 3, pp. 355–395. Oxford University Press, Oxford (1994)
5. DeLP Web page, http://lidia.cs.uns.edu.ar/delp
6. Pollock, J.L.: Knowledge and Representation. Princeton University Press, Princeton (1974)
7. Pollock, J.L.: Defeasible reasoning. Cognitive Science 11(4), 481–518 (1987)
8. Toulmin, S.E.: The Uses of Argument. Cambridge University Press, Cambridge (1958)
9. Verheij, B.: Deflog: On the logical interpretation of prima facie justified assumptions. Journal of Logic and Computation 13(3), 319–346 (2003)
10. Verheij, B.: Dialectical argumentation with argumentation schemes: An approach to legal logic. Artificial Intelligence and Law 11(2-3), 167–195 (2003)
11. Verheij, B.: Evaluating arguments based on toulmin's scheme. Argumentation 19(3), 347–371 (2005)

An Argumentation Machinery to Reason over Inconsistent Ontologies*

Martín O. Moguillansky[1], Renata Wassermann[2], and Marcelo A. Falappa[1]

[1] CONICET, AI R&D Lab (LIDIA), Universidad Nacional del Sur (UNS), Argentina
{mom,maf}@cs.uns.edu.ar
[2] Department of Computer Science (DCC), University of São Paulo (USP), Brazil
renata@ime.usp.br

Abstract. Widely accepted argumentation techniques are adapted to define a non-standard description logic (DL) reasoning machinery. A DL-based argumentation framework is introduced to reason about potentially inconsistent ontologies. Arguments in this framework can handle different DL families like $\mathcal{ALC}$, $\mathcal{EL}$, and DL-Lite. Afterwards, we propose an algorithm based on debugging techniques and classical tableau-based $\mathcal{ALC}$ satisfiability to build arguments, and discuss about the computational cost of reasoning through the proposed machinery.

1 Introduction

Much effort has been dedicated to the study of models for debugging and repairing ontologies, aiming at restoring consistency. One of the most influencing works on this matter is Schlobach and Cornet's [12]. However, in ontologies conceptualizing certain domains like medicine and law, it is well known the need to avoid loosing beliefs disregarding inconsistencies. In such areas, paraconsistent methods are usually needed in order to reason over inconsistency. Argumentation theory appears as an interesting alternative. For instance, in [10], ontologies (with some restrictions) are translated into the defeasible logic programming argumentation system (DeLP) [9]. Nonetheless, it is more desirable to benefit from the continuous advances done in the area of ontology reasoning, and thus building DL-argumentation machineries on top of the corresponding specialized DL reasoner. This would also avoid any unnecessary additional complexity arising from the translation of ontologies to different logics. The main objective of this work is focused on the study of such DL-argumentation theories.

Some argumentation systems have been implemented with promisory results, as is the case of DeLP among others. However, the study of complexity of reasoning through argumentation is still at its initial stages. Some results are available in [6]. In this article, we walk our first steps towards the complexity analysis of our proposed DL-argumentation machinery, which is defined upon DL semantic entailment. This allows to reason by relying on interpretation models, and therefore, tableaux techniques may be easily reused towards future implementations. Hence, since this methodology can be implemented on top of the corresponding DL reasoner, its complexity will be attached to that of the problem of reasoning in the underlying DL. We show this on ontologies

* Partially supported by UNS (PGI 24/ZN18) and CONICET (PIP 112-200801-02798).

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 100–109, 2010.

based on an $\mathcal{ALC}$ fragment, known as unfolded $\mathcal{ALC}$, through an algorithm for building the set of defeaters of a given argument. Such algorithm is shown to correspond to the complexity class of the problem of reasoning in unfolded $\mathcal{ALC}$, which is in PSPACE. This algorithm would be useful for building dialectical trees (trees of arguments), as the elementary structure upon which the adopted argumentation semantics rely.

2 Description Logics: Brief Overview

An ontology $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ details knowledge in terms of intensional/extensional information described in $\mathcal{T}$ (TBox)/$\mathcal{A}$ (ABox). The symbols A and C identify atomic and general concepts, and P and E, atomic and general roles, respectively. Symbols $a, b, \ldots$ identify constant named individuals from the domain, and $x, y, \ldots$, free variables. The sets $\mathrm{N_V}$ and $\mathrm{N_C}$ are assumed to contain variables and constant names, respectively. A TBox usually contains axioms like $C_1 \sqsubseteq C_2$ identified as *general concept inclusions (GCIs)*; and an ABox, *membership assertions* like $C(a)$ and $E(a, b)$. However, different restrictions may appear depending on each specific description language. From now on we assume the reader is familiar with DLs. For more details on DLs refer to [2].

DL-semantics is given in terms of the standard set theoretic Tarskian semantics, through *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. If the interpretation $\mathcal{I}$ is model of the ontology Σ, we write $\mathcal{I} \models \Sigma$, implying $\mathcal{I} \models \phi$, for every assertion $\phi \in \Sigma$. An ontology is *satisfiable* (or *consistent*) if it admits at least one model. An ontology is *coherent* if it is satisfiable and for every concept C and every model $\mathcal{I}$, $C^{\mathcal{I}}$ is non-empty. An *incoherent* ontology considers a concept C which does not accept any individual in some model; however satisfiability could still hold in such ontology. Finally, an ontology Σ *logically implies* (or *entails*) an assertion ϕ, written $\Sigma \models \phi$, if for every model $\mathcal{I}$ of Σ, $\mathcal{I} \models \phi$.

To verify logical implications of DL formulae we rely on *reasoning services (RS)* like *subsumption* ($\Sigma \models C_1 \sqsubseteq C_2$), *instance checking* ($\Sigma \models C(a)$ or $\Sigma \models E(a, b)$), and *knowledge base satisfiability* (whether the ontology admits at least one model). In addition, *query answering* ($\Sigma \models q(\bar{x})$) has been most widely considered for querying DLs. A *conjunctive query (cq)* $q(\bar{x})$, with a tuple $\bar{x} \in (\mathrm{N_V})^n$ of arity $n \geq 0$, is a non empty set of atoms $C(z)$, $E(z_1, z_2)$, $z_1 = z_2$, or $z_1 \neq z_2$, where C and E are respectively a general concept and a general role of Σ, and only the names $\{z, z_1, z_2\} \cap \mathrm{N_V}$ are considered in $\bar{x}$. Function $\mathfrak{var} : (\mathrm{N_V})^n \longrightarrow 2^{\mathrm{N_V}}$ identifies the variables in $\bar{x}$. When $\bar{x}$ is the empty tuple, no free variables are considered and the query is identified as *boolean*. Intuitively, $q(\bar{x})$ represents the conjunction of its elements. Let $\mathcal{I}$ be an interpretation, and $\mathfrak{m} : \mathfrak{var}(\bar{x}) \cup \mathrm{N_C} \longrightarrow \Delta^{\mathcal{I}}$ a total function. If $z \in \mathrm{N_C}$ then $\mathfrak{m}(z) = z^{\mathcal{I}}$ otherwise $\mathfrak{m}(z) = a \in \Delta^{\mathcal{I}}$. We write $\mathcal{I} \models^{\mathfrak{m}} C(z)$ if $\mathfrak{m}(z) \in C^{\mathcal{I}}$, $\mathcal{I} \models^{\mathfrak{m}} E(z_1, z_2)$ if $(\mathfrak{m}(z_1), \mathfrak{m}(z_2)) \in E^{\mathcal{I}}$, $\mathcal{I} \models^{\mathfrak{m}} (z_1 = z_2)$ if $\mathfrak{m}(z_1) = \mathfrak{m}(z_2)$, and $\mathcal{I} \models^{\mathfrak{m}} (z_1 \neq z_2)$ if $\mathfrak{m}(z_1) \neq \mathfrak{m}(z_2)$. If $\mathcal{I} \models^{\mathfrak{m}} \phi$ for all $\phi \in q(\bar{x})$, we write $\mathcal{I} \models^{\mathfrak{m}} q(\bar{x})$ and call $\mathfrak{m}$ a *match* for $\mathcal{I}$ and $q(\bar{x})$. We say that $\mathcal{I}$ satisfies $q(\bar{x})$ and write $\mathcal{I} \models q(\bar{x})$ if there is a match $\mathfrak{m}$ for $\mathcal{I}$ and $q(\bar{x})$. If $\mathcal{I} \models q(\bar{x})$ for all models $\mathcal{I}$ of an ontology Σ, we write $\Sigma \models q(\bar{x})$ and say that Σ entails $q(\bar{x})$. Note that by answering cqs, instance checking ends up subsumed.

Reasoning in the standard $\mathcal{ALC}$ ($C \longrightarrow A | \bot | \top | \neg C | C \sqcap C | C \sqcup C | \forall P.C | \exists P.C$) is EXPTIME-complete, however, by restricting their use to unfoldable $\mathcal{ALC}$, satisfiability turns to PSPACE-complete. (A TBox is called unfoldable if the left-hand sides of axioms (defined concepts) are atomic, and if the right-hand sides (definitions) contain no

direct or indirect reference to the defined concept.) The panorama gets worse by relying upon more expressive DLs. Different DLs have been proposed aiming at reducing the complexity of reasoning in detriment of their expressivity. That is the case of $\mathcal{EL}$ [1] ($C \longrightarrow A|\bot|\top|C \sqcap C|\exists P.C$), whose satisfiability checking was shown polinomial.

For $\mathcal{ALC}$ (and some extensions) and $\mathcal{EL}$, we will only consider the RSs of *subsumption* $\Sigma \models C_1 \sqsubseteq C_2$; and *query answering* $\Sigma \models q(\bar{x})$. In addition, for those languages whose names contain the letter $\mathcal{H}$, subsumption also corresponds to $\Sigma \models E_1 \sqsubseteq E_2$. This is the case of $\mathcal{ALCH}$, $\mathcal{ELH}$, and $\mathcal{ALCNH}^{-1,\neg}$ (see Ex. 5), among others.

A particular family of DLs which aims at lower complexity in detriment of expressivity is DL-Lite [5]. These logics allow to reason about large amounts of assertional data (ABox). Data complexity of query answering (wrt. the size of the ABox) is in LOGSPACE for most of the DL-Lite members, and polinomial regarding the whole ontology. Moreover, queries over DL-Lite ontologies may be rewritten as SQL queries so that standard database query engines can be used. The *DL-Lite*$_{\texttt{core}}$ grammar is given by $B \longrightarrow A|\exists R$, $C \longrightarrow B|\neg B$, $R \longrightarrow P|P^-$, and $E \longrightarrow R|\neg R$. In *DL-Lite*$_{\texttt{core}}$, the TBox is formed by axioms like $B \sqsubseteq C$, and the ABox by membership assertions like $A(a)$ and $P(a,b)$. Adding to *DL-Lite*$_{\texttt{core}}$ axioms like $R \sqsubseteq E$, or *functional restrictions* like $(\mathrm{funct}R)$, renders the languages *DL-Lite*$_{\mathcal{R}}$ and *DL-Lite*$_{\mathcal{F}}$, respectively. (An interpretation $\mathcal{I}$ is a model of a $(\mathrm{funct}R)$ if the binary relation $R^{\mathcal{I}}$ is a function, *i.e.*, $(x, y_1) \in R^{\mathcal{I}}$ and $(x, y_2) \in R^{\mathcal{I}}$ implies $y_1 = y_2$. In this case, the *functionality checking* RS $\Sigma \models (\mathrm{funct}R)$ and $\Sigma \models \neg(\mathrm{funct}R)$, is considered.) Combining both extensions renders *DL-Lite*$_{(\mathcal{RF})}$, whose satisfiability turns to EXPTIME-hard.

3 DL Argumentation

An *argument* may be seen as a set of interrelated pieces of knowledge (in a language $\mathcal{L}$) providing support to a claim. A language $\mathcal{L}_{\texttt{cl}}$ for claims is assumed: for any $\phi \in \mathcal{L}_{\texttt{cl}}$, there is a set $\Phi \subseteq \mathcal{L}$ such that $\Phi \models \phi$. Caligraphic letters $\mathcal{A}$, $\mathcal{B}$, ..., denote arguments.

Definition 1 (Argument). *An **argument** $\mathcal{B}$ is a structure $\langle \Delta, \beta \rangle$, where $\Delta \subseteq \mathcal{L}$ is the **body**, $\beta \in \mathcal{L}_{\texttt{cl}}$ the **claim**, and it holds* (1) $\Delta \models \beta$, (2) *Δ is satisfiable (or $\Delta \not\models \bot$), and* (3) $\nexists X \subset \Delta : X \models \beta$. *We say that $\mathcal{B}$ **supports** β.*

Queries done to a knowledge base (KB) are supported through the claim of arguments. For ontology reasoning, we assume arguments' bodies to be included in an ontology $\Sigma \subseteq \mathcal{L}$, where $\mathcal{L}$ is the underlying DL in Σ. In this work, we write $\mathcal{L}$ to refer to general description languages (unless we reify $\mathcal{L}$ to a specific DL).

The *domain of arguments from an ontology* Σ is identified through the set $\mathbb{A}_\Sigma$. By means of $\mathfrak{bd} : \mathbb{A}_\Sigma \longrightarrow 2^{\mathcal{L}}$ and $\mathfrak{cl} : \mathbb{A}_\Sigma \longrightarrow \mathcal{L}_{\texttt{cl}}$, the body $\mathfrak{bd}(\mathcal{B})$ and claim $\mathfrak{cl}(\mathcal{B})$ of an argument $\mathcal{B} \in \mathbb{A}_\Sigma$, may be respectively identified. Observe that claims (within $\mathcal{L}_{\texttt{cl}}$) conform to $\mathcal{L}$ since they are not necessarily contained in Σ but entailed by Σ. The entailment $\models$ is obtained through usual DL interpretations, and for querying ontologies we refer to RSs as presented before. Hence, arguments should support RSs through their claims conforming the language $\mathcal{L}_{\texttt{cl}} \longrightarrow C_1 \sqsubseteq C_2|q(\langle\rangle)$, and we additionally consider $E_1 \sqsubseteq E_2$, when $\mathcal{L} =$ *DL-Lite*$_{\mathcal{R}}$ or the $\mathcal{L}$ string contains an $\mathcal{H}$; and $(\mathrm{funct}R)|\neg(\mathrm{funct}R)$, when $\mathcal{L} =$ *DL-Lite*$_{(\mathcal{RF})}$ or *DL-Lite*$_{\mathcal{F}}$. Observe that from $\mathcal{L}_{\texttt{cl}}$'s

syntax, $q(\langle\rangle)$ refers to a cq with an empty tuple from $\mathrm{N_V}$. Hence, no argument supports a claim with free variables. That is, for any ontology $\Sigma \subseteq \mathcal{L}$ and any argument $\mathcal{B} \in \mathbb{A}_\Sigma$, if $\mathfrak{cl}(\mathcal{B}) = q(\bar{x})$ then $\bar{x} = \langle\rangle$. Given a query $\alpha \in \mathcal{L}_{\mathfrak{cl}}$, an argument $\mathcal{B}$ is a *query supporter* if $\mathcal{B}$ supports α. We define the notion of query supporter to cope with any RS.

Definition 2 (Query Supporter). *Given an ontology $\Sigma \subseteq \mathcal{L}$ and a query $\alpha \in \mathcal{L}_{\mathfrak{cl}} \cup \{q(\bar{x})\}$; an argument $\mathcal{B} \in \mathbb{A}_\Sigma$ is an α-**supporter** iff for every model $\mathcal{I}$ of $\mathfrak{cl}(\mathcal{B})$ there exists a match $\mathfrak{m}$ for $\mathcal{I}$ and α such that $\mathcal{I} \models^{\mathfrak{m}} \alpha$ holds, or equivalently $\mathfrak{cl}(\mathcal{B}) \models \alpha$.*

Primitive arguments like $\langle\{A(a)\}, A(a)\rangle$ appear when $A(a) \in \Sigma$; assuming $\{A(a), A \sqsubseteq B\} \subseteq \Sigma$, more complex arguments appear within $\mathbb{A}_\Sigma$, like $\langle\{A \sqsubseteq B\}, A \sqsubseteq B\rangle$ and $\langle\{A(a), A \sqsubseteq B\}, B(a)\rangle$. An argument $\mathcal{B} \in \mathbb{A}_\Sigma$ is *subargument* of $\mathcal{C} \in \mathbb{A}_\Sigma$ (and conversely, $\mathcal{C}$ is a *superargument* of $\mathcal{B}$) if it follows that $\mathfrak{bd}(\mathcal{B}) \subseteq \mathfrak{bd}(\mathcal{C})$ holds; moreover, $\mathcal{B}$ is a *proper subargument* of $\mathcal{C}$ if $\mathfrak{bd}(\mathcal{B}) \subset \mathfrak{bd}(\mathcal{C})$ holds. Observe that $\langle\{A \sqsubseteq B\}, A \sqsubseteq B\rangle$ is a subargument of $\langle\{A(a), A \sqsubseteq B\}, B(a)\rangle$. A functionality checking $(\mathrm{funct}R)$ is supported through $\langle\{(\mathrm{funct}R)\}, (\mathrm{funct}R)\rangle$. For other arguments considering functional assertions, assume for instance, $\neg P(a, c)$ is supported through both $\langle\{P(a, b), (\mathrm{funct}P)\}, \{\neg P(a, c)\}\rangle$ and $\langle\{P(b, c), (\mathrm{funct}P^-)\}, \{\neg P(a, c)\}\rangle$. CQ's like $q(\bar{x})$, where for instance $q(\bar{x}) = \{A(x), B(y)\}$ and $\bar{x} = \langle x, y\rangle$, might be supported through an argument $\mathcal{B} = \langle\{A(a), A \sqsubseteq B\}, q(\bar{y})\rangle$, where $q(\bar{y}) = \{A(a), B(a)\}$, $\bar{y} = \langle\rangle$, and a match $\mathfrak{m}(x) = \mathfrak{m}(y) = a$, appears. Observe that by changing the query to $q(\bar{x}) = \{A(x), B(y), x \neq y\}$, argument $\mathcal{B}$ is no longer a $q(\bar{x})$-supporter. From now on we refer to arguments like $\mathcal{B}$ as $\langle\{A(a), A \sqsubseteq B\}, \{A(a), B(a)\}\rangle$, avoiding the explicit writing of cq's like $q(\langle\rangle) = \{A(a), B(a)\}$ as claims.

Counterarguments are arguments whose claims pose justifications to disbelieve in other arguments. From the standpoint of an ontology Σ, counterarguments bring about sources of incoherence/inconsistency determined by their bodies. Counterarguments are usually formalized upon negation of formulae, nonetheless, this is problematic in DLs since negation of axioms can fall beyond the scope of the language. Hence, counterarguments are formalized upon DL satisfiability, avoiding the use of negation of axioms.

Definition 3 (Counterargument). *Given $\Sigma \subseteq \mathcal{L}$, arguments $\mathcal{B} \in \mathbb{A}_\Sigma$ and $\mathcal{C} \in \mathbb{A}_\Sigma$ are **in conflict** iff $\mathfrak{bd}(\mathcal{B}) \cup \mathfrak{cl}(\mathcal{C})$ is unsatisfiable. Argument $\mathcal{C}$ is a **counterargument** of $\mathcal{B}$.*

Argument-based preference relations usually determine *defeats* between arguments. However, for simplicity we avoid its usage, and identify *defeaters* strightforwardly from counterarguments: $\mathcal{C}$ *defeats* $\mathcal{B}$ (noted $\mathcal{C} \hookrightarrow \mathcal{B}$) *iff* $\mathcal{C}$ counterargues $\mathcal{B}$.

Usually, arguments can be *rebutted* with an argument of opposite conclusion and they can be *undercut* with an argument whose conclusion opposes either explicitly or implicitly to the body of the defeated argument. Given two arguments $\mathcal{B}$ and $\mathcal{C}$ such that $\mathcal{C}$ counterargues $\mathcal{B}$, we usually say that $\mathcal{C}$ rebutts $\mathcal{B}$ if it follows that $\mathfrak{cl}(\mathcal{B}) = \neg\mathfrak{cl}(\mathcal{C})$. Nevertheless, for the case of DL-arguments this may not be so trivial. Let us analyze with more detail the negation of axioms in the context of DL arguments. Given an argument $\mathcal{B} = \langle\{A \sqsubseteq B, B \sqsubseteq C\}, A \sqsubseteq C\rangle$, a possible rebuttal would support an axiom like $\neg(A \sqsubseteq C)$. Negation of axioms like $A \sqsubseteq C$ was studied in [8] determining two kinds of negation: *consistency-negation* $\neg(A \sqsubseteq C) = \exists(A \sqcap \neg C)$ and *coherency-negation* $\sim(A \sqsubseteq C) = A \sqsubseteq \neg C$. For the former, an existence assertion like $\exists(A \sqcap \neg C)(x)$ would serve, however for languages without concept conjunction like *DL-Lite*$_{(\mathcal{RF})}$,

it would fall out of the scope. For such cases, the existence assertion can be rewritten as a cq like $q(\langle x \rangle) = \{A(x), \neg C(x)\}$. For instance, $\mathcal{C} = \langle \{A(a), A \sqsubseteq \neg C\}, \{A(a), \neg C(a)\} \rangle$ would be a possible rebuttal supporting $q(\langle x \rangle)$ with $\mathfrak{m}(x) = a$. On the other hand, coherency-negation may be simply achieved by looking for an argument supporting $A \sqsubseteq \neg C$. In addition, for *DL-Lite*$_{\mathcal{F}}$, we extend negation of axioms to functional assertions, interpreting $\neg(\mathrm{funct} R)$ as a role R that does not conform to the definition of a function. An argument supporting such negation should consider extensional information (ABox). For instance, $\langle \{P(a,b), P'(a,c), P' \sqsubseteq P\}, \neg(\mathrm{funct} P) \rangle$.

Nonetheless, a major drawback appears when considering query answering: negation of cqs is in general undefined. As an alternative solution, we concentrate on finding sources of DL unsatisfiability through *undercuts*: $\mathcal{C}$ undercuts $\mathcal{B}$ *iff* $\mathcal{C}$ counterargues $\mathcal{B}$. Several undercuts may appear for a same argument. Furthermore, as is shown next, an undercut of an argument $\mathcal{B}$ may encompass other defeaters of $\mathcal{B}$.

Example 1. Consider the *DL-Lite*$_{\mathtt{core}}$ arguments $\mathcal{B} = \langle \{A(a), A \sqsubseteq B\}, \{B(a)\} \rangle$ and $\mathcal{C} = \langle \{A(b), A \sqsubseteq C, C \sqsubseteq \neg B\}, \{A(b), \neg B(b)\} \rangle$, and $\mathcal{C}$'s subargument $\mathcal{C}' = \langle \{A \sqsubseteq C, C \sqsubseteq \neg B\}, A \sqsubseteq \neg B \rangle$. Both $\mathcal{C}$ and $\mathcal{C}'$ counterargue $\mathcal{B}$.

Canonical undercuts [4] were defined (upon classic logic) as representatives of all defeaters of an argument. The claim of a canonical undercut $\mathcal{C}$ negates the conjunctive enumeration of all beliefs from the body of the counterargued argument $\mathcal{B}$, *i.e.*, $\mathfrak{cl}(\mathcal{C}) = \neg(\alpha_1 \wedge \ldots \wedge \alpha_n)$, where $\mathfrak{bd}(\mathcal{B}) = \{\alpha_1, \ldots, \alpha_n\}$. However, such kind of claims would fall out of the scope of a DL $\mathcal{L}$. Canonical undercuts constitute minimal sources of inconsistency (taken from the KB they are built) wrt. the body of the counterargued argument. Thus, by following such intuition, we propose *minimal undercuts*.

Definition 4 (Minimal Undercut). *Let $\mathcal{B}, \mathcal{C} \in \mathbb{A}_{\Sigma}$ be such that $\mathcal{C}$ counterargues $\mathcal{B}$. $\mathcal{C}$ is a **minimal undercut** of $\mathcal{B}$ iff there is no proper subargument of $\mathcal{C}$ counterarguing $\mathcal{B}$.*

To illustrate this notion, note from Ex. 1, that $\mathcal{C}'$ is a minimal undercut of $\mathcal{B}$. Through the use of minimal undercuts, we have restricted the consideration of counterarguments to those of minimal body. However, for some DLs several minimal undercuts with different claims may appear (see Ex. 2). In such cases, we will keep those of maximum entailment, identified as *maximally conservative undercuts* (*mcu*, for short).

Definition 5 (Maximally Conservative Undercut). *Let $\mathcal{B} \in \mathbb{A}_{\Sigma}$ and $\mathcal{C} \in \mathbb{A}_{\Sigma}$ be such that $\mathcal{C}$ minimally undercuts $\mathcal{B}$. Argument $\mathcal{C}$ is a **maximally conservative undercut (mcu)** of $\mathcal{B}$ iff for every subargument $\mathcal{C}'$ of $\mathcal{C}$ minimally undercutting $\mathcal{B}$, $\mathfrak{cl}(\mathcal{C}) \models \mathfrak{cl}(\mathcal{C}')$ holds.*

Example 2. Let $\mathcal{C} = \langle \Psi, A \sqcup B \sqsubseteq \neg C \rangle$, $\mathcal{C}' = \langle \Psi, A \sqsubseteq \neg C \rangle$, and $\mathcal{C}'' = \langle \Psi, B \sqsubseteq \neg C \rangle$, with $\Psi = \{A \sqcup B \sqsubseteq D, D \sqsubseteq \neg C\}$; be three $\mathcal{ALC}$ minimal undercuts of $\mathcal{B} = \langle \{(A \sqcap B)(a), A \sqcup B \sqsubseteq C\}, \{(A \sqcup B)(a), C(a)\} \rangle$. Note that only $\mathcal{C}$ is an mcu of $\mathcal{B}$.

Classic argumentation semantics like those from [7] (and others) can be applied to the DL-argumentation framework presented here. Nonetheless, ontology reasoning requires practical approaches. Hence, we should control the number of arguments needed to reason. Since querying ontologies can be performed via RSs (Sect. 2), the reasoning methodology we assume is based on the acceptability of some query supporter obtained from $\mathbb{A}_{\Sigma}$. Thus, we rely on an argumentation semantics defined upon *dialectical trees*.

One of the essential elements of this semantics is the notion of *argumentation line*: a non-empty sequence $\lambda = [\mathcal{B}_1 \ldots, \mathcal{B}_n]$ of arguments from $\mathbb{A}_\Sigma$, where $\mathcal{B}_i \hookrightarrow \mathcal{B}_{i-1}$, for any $1 < i \leq n$. Argument $\mathcal{B}_1$ is identified as λ's root, and $\mathcal{B}_n$, as λ's leaf. An argumentation line could be seen as *two parties engaged in a discussion*: one standing by the root argument and the other arguing against it. Consequently, given a line λ, we identify the *set of pro (resp, con) arguments* containing all arguments placed on odd (resp, even) positions in λ, noted as λ^+ (resp, λ^-). We will abuse notation and write $\mathcal{B} \in \lambda$ to identify $\mathcal{B}$ from the argumentation line λ. An initial sequence of arguments in a line $\lambda = [\mathcal{B}_1, \ldots, \mathcal{B}_n]$ is identified through its *upper segment* $\lambda^{\uparrow}[\mathcal{B}_i] = [\mathcal{B}_1, \ldots, \mathcal{B}_i]$, with $1 \leq i \leq n$. Besides, the *proper upper segment* of λ wrt. $\mathcal{B}_i$ $(i \neq 1)$ is defined as $\lambda^{\uparrow}(\mathcal{B}_i) = [\mathcal{B}_1, \ldots, \mathcal{B}_{i-1}]$. We refer to both proper and non-proper upper segments simply as "upper segments" and will be distinguishable only through the notation (round or square brackets respectively). *Acceptability conditions* are used to determine finite and non-fallacious lines, named *acceptable argumentation lines*, which are ensured to be non-circular and concordant. A line $[\mathcal{B}_1, \ldots, \mathcal{B}_n]$ is *non-circular iff* for any $1 \leq i \leq n$, the smallest subargument $\mathcal{C}$ of $\mathcal{B}_i$ such that $\mathcal{C}$ is defeated by $\mathcal{B}_{i+1}$, is not reintroduced in the rest of the line, that is $\mathcal{C}$ is not a subargument of any argument $\mathcal{B}_j$, with $i < j \leq n$. A line λ is *concordant iff* the sets $\bigcup_{\mathcal{B} \in \lambda^+} \mathfrak{bd}(\mathcal{B})$ and $\bigcup_{\mathcal{B} \in \lambda^-} \mathfrak{bd}(\mathcal{B})$ of bodies of pro and con arguments (respectively) are individually satisfiable. More on acceptability conditions can be found in [9]. Additionally, lines will be required to be *exhaustive*: if λ is acceptable, extending it with any defeater of its leaf determines a non-acceptable line.

On the other hand, the acceptability of a query supporter $\mathcal{R} \in \mathbb{A}_\Sigma$, namely the *warrant* status of $\mathcal{R}$, is determined by analizing the *dialectical tree* rooted in $\mathcal{R}$. Such tree is built from a maximal set of argumentation lines rooted in $\mathcal{R}$. Reducing the number of arguments' defeaters allows us to shrink the set of argumentation lines used to build dialectical trees. The usage of mcus results benefitting to such end, allowing to consider a single defeater from a set of minimal undercuts of a common argument. However, for DLs where logic equivalence of claims is possible, several mcus may appear.

Example 3. Given $\Sigma \subseteq \mathcal{ALC}$ and $\mathcal{B} = \langle \{A \sqsubseteq B', B' \sqsubseteq \neg B\}, A \sqsubseteq \neg B \rangle$ in $\mathbb{A}_\Sigma$. Assuming $\langle \{A(a), B(a)\}, \{(A \sqcap B)(a)\} \rangle$ and $\langle \{A(a), B(a)\}, \{A(a), B(a)\} \rangle$ can be built, both arguments are mcus of $\mathcal{B}$ with identical bodies and logically equivalent claims.

Building dialectical trees requires to identify *bundle sets*: maximal sets of argumentation lines rooted in a common argument. To such end, we define a domain $\mathbb{L}_\Sigma$ containing all the acceptable and exhaustive lines. $\mathbb{L}_\Sigma$ is ensured to be free of redundancies as seen in Ex. 3, by restricting the use of mcus with equivalent claims (non-redundancy).

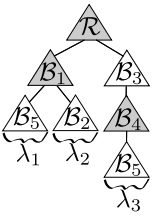
Definition 6 (Argumentation Line Domain). *Given an ontology Σ, the **argumentation line domain** $\mathbb{L}_\Sigma$, is the maximal set of argumentation lines $\lambda = [\mathcal{B}_1, \ldots, \mathcal{B}_n]$, where $\mathcal{B}_i \in \mathbb{A}_\Sigma$ (for $1 \leq i \leq n$), such that λ is acceptable, exhaustive, **maximally conservative** ($\mathcal{B}_j$ is an mcu of $\mathcal{B}_{j-1}$ (for $1 < j \leq n$)), and **non-redundancy** is guaranteed:*

(non-redundancy) *for any $\{\lambda, \lambda'\} \subseteq \mathbb{L}_\Sigma$, $\mathcal{B} \in \lambda$ and $\mathcal{C} \in \lambda'$, if $\mathfrak{bd}(\mathcal{B}) = \mathfrak{bd}(\mathcal{C})$ and $\lambda^{\uparrow}(\mathcal{B}) = \lambda'^{\uparrow}(\mathcal{C})$ then $\lambda = \lambda'$.*

A *bundle set* for $\mathcal{R}$, noted as $\mathcal{S}(\mathcal{R})$, contains all the argumentation lines from $\mathbb{L}_\Sigma$ rooted in $\mathcal{R}$. From a bundle set $\mathcal{S}(\mathcal{R})$, the dialectical tree $\mathcal{T}(\mathcal{R})$ is constructed.

Definition 7 (Dialectical Tree). *Given an ontology $\Sigma \subseteq \mathcal{L}$, a **dialectical tree** $\mathcal{T}(\mathcal{R})$ rooted in $\mathcal{R} \in \mathbb{A}_\Sigma$ is determined by the bundle set $\mathcal{S}(\mathcal{R}) \subseteq \mathbb{L}_\Sigma$ such that an argument $\mathcal{C}$ in $\mathcal{T}(\mathcal{R})$ is: (1) a **node** iff $\mathcal{C} \in \lambda$, for any $\lambda \in \mathcal{S}(\mathcal{R})$; (2) a **child** of a node $\mathcal{B}$ in $\mathcal{T}(\mathcal{R})$ iff $\mathcal{C} \in \lambda$, $\mathcal{B} \in \lambda'$, for any $\{\lambda, \lambda'\} \subseteq \mathcal{S}(\mathcal{R})$, and $\lambda'^{\uparrow}[\mathcal{B}] = \lambda^{\uparrow}(\mathcal{C})$. The **leaves** in $\mathcal{T}(\mathcal{R})$ are the leaves of each line in $\mathcal{S}(\mathcal{R})$. The domain of trees from Σ is noted as $\mathbb{T}_\Sigma$.*

Example 4.
Given an ontology $\Sigma \subseteq \mathcal{L}$, and the bundle set $\mathcal{S}(\mathcal{R}) \subseteq \mathbb{L}_\Sigma$ determining the dialectical tree $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$ (depicted on the right) such that $\mathcal{S}(\mathcal{R}) = \{\lambda_1, \lambda_2, \lambda_3\}$, where $\lambda_1 = [\mathcal{R}, \mathcal{B}_1, \mathcal{B}_5]$, $\lambda_2 = [\mathcal{R}, \mathcal{B}_1, \mathcal{B}_2]$, and $\lambda_3 = [\mathcal{R}, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5]$, are three acceptable, maximally conservative, and non-redundant lines. Observe that argument $\mathcal{B}_2$ is a child of $\mathcal{B}_1$ in $\mathcal{T}(\mathcal{R})$ given that $\lambda_1^{\uparrow}[\mathcal{B}_1] = \lambda_2^{\uparrow}(\mathcal{B}_2)$ (see Def. 7).



We write $\lambda \in \mathcal{T}(\mathcal{R})$ when λ is a line in $\mathcal{T}(\mathcal{R})$. Given $\Sigma \subseteq \mathcal{L}$, a query supporter $\mathcal{R} \in \mathbb{A}_\Sigma$ is finally accepted (or warranted) by analizing the dialectical tree $\mathcal{T}(\mathcal{R})$. To such end, a *marking function* $\mathtt{mark} : \mathbb{A}_\Sigma \times \mathbb{L}_\Sigma \times \mathbb{T}_\Sigma \longrightarrow \mathbb{M}$ assigns to each argument in $\mathcal{T}(\mathcal{R})$ a mark from $\mathbb{M} = \{D, U\}$, where *D/U* means defeated/undefeated. The mark of an inner node in $\mathcal{T}(\mathcal{R})$ is obtained from its children (*i.e.*, its defeaters) by following a *marking criterion*. We adopt a skeptical marking criterion (as used in DeLP [9]) defined as: (1) all leaves are marked U and (2) every inner node $\mathcal{B}$ is marked U *iff* every child of $\mathcal{B}$ is marked D, otherwise, $\mathcal{B}$ is marked D. The *warranting function* $\mathtt{warrant} : \mathbb{T}_\Sigma \longrightarrow \{true, false\}$ determines the root's acceptance verifying $\mathtt{warrant}(\mathcal{T}(\mathcal{R})) = true$ *iff* $\mathtt{mark}(\mathcal{R}, \lambda, \mathcal{T}(\mathcal{R})) = U$. Hence, $\mathcal{R}$ is *warranted* from $\mathcal{T}(\mathcal{R})$ *iff* $\mathtt{warrant}(\mathcal{T}(\mathcal{R})) = true$. In such a case, $\mathcal{T}(\mathcal{R})$ is referred as *warranting tree*. These notions are illustrated with arguments painted in grey/white standing for *D/U* marks. For instance, in Ex. 4, $\mathcal{R}$ is defeated and thus $\mathcal{T}(\mathcal{R})$ is non-warranting.

Definition 8 (Argumentative Entailment). *Given $\Sigma \subseteq \mathcal{L}$ and a query $\alpha \in \mathcal{L}_{\mathtt{cl}} \cup \{q(\bar{x})\}$; $\Sigma \mid\!\approx \alpha$ iff there is a warranted α-supporter $\mathcal{R} \in \mathbb{A}_\Sigma$. If there is no warranted α-supporter from $\mathbb{A}_\Sigma$ then α is not entailed by Σ, noted as $\Sigma \not\mid\!\approx \alpha$.*

Theorem 1. *Given an ontology $\Sigma \subseteq \mathcal{L}$, if Σ is coherent and consistent then for any query $\alpha \in \mathcal{L}_{\mathtt{cl}} \cup \{q(\bar{x})\}$ it holds $\Sigma \models \alpha$ iff $\Sigma \mid\!\approx \alpha$.*

Example 5. To analyze whether a given presidential formula is reliable, we study how candidates voted for the last most relevant laws in the parlament, deciding in this manner whether a pair of candidates might be coalitionable. For that matter, we consider a role P, standing for presidential formula such that for any $(x, y) \in P^{\mathcal{I}}$, individual x is candidate for president and y for vice-president; role C (coalitionable candidates) such that for any $(x, y) \in C^{\mathcal{I}}$, individual x and y are two politicians that are assumed to agree according to their ideology on the most important national matters; concepts L and L', standing for the two most relevant laws being promulged during the last presidential period such that $a \in L^{\mathcal{I}}$ (resp., $a \in L'^{\mathcal{I}}$) identifies the politician that voted in favor of the L's (resp, L''s) promulgation; concept L_1, standing for one of the most controversial articles from L's promulgation such that $a \in L_1^{\mathcal{I}}$ identifies the politician that voted in favor of L_1. The ontology $\Sigma \subseteq \mathcal{ALCNH}^{-1,\neg}$ will contain axioms $P \sqsubseteq C$ (every

presidential formula is coalitionable); $C \sqsubseteq C^-$ (every coalitionable pair of politicians is commutative); $L_1 \sqsubseteq L$ (politicians in favor of article L_1 should have voted in favor of law L); $\forall P.\top \sqsubseteq= 1P$ and $\forall P^-.\top \sqsubseteq= 1P^-$ (a presidential formula should be unique and the candidates should have the expected position explicitly announced, and candidates presented in several presidential formulas are assumed to be less reliable); and $\forall C.L \sqsubseteq L$ and $\forall C.L' \sqsubseteq L'$ (politicians agreeing in L or L' are coalitionable). The Σ's ABox will include $P(a,b)$, $P(a,c)$, $P(d,e)$, $\neg C(d,e)$, $L(a)$, $\neg L(b)$, $L_1(b)$, $\neg L'(a)$, $L'(c)$, where individuals a, b, c, d, e, are currently active politicians.

Checking the reliability of the presidential formula $P(a,b)$ implies finding out whether $\Sigma \mid\!\approx P(a,b)$ holds. From $\mathbb{A}_\Sigma$, $\mathcal{R} = \langle\{P(a,b)\}, \{P(a,b)\}\rangle$ is a query supporter, and $\mathcal{B}_1 = \langle\{P \sqsubseteq C, C \sqsubseteq C^-, \forall C.L \sqsubseteq L, L(a), \neg L(b)\}, \{\neg P(a,b)\}\rangle$, $\mathcal{B}_2 = \langle\{L_1(b), L_1 \sqsubseteq L\}, \{L(b)\}\rangle$, $\mathcal{B}_3 = \langle\{P(a,c), \forall P.\top \sqsubseteq= 1P\}, \{\neg P(a,b)\}\rangle$, $\mathcal{B}_4 = \langle\{P \sqsubseteq C, \forall C.L' \sqsubseteq L', \neg L'(a), L'(c)\}, \{\neg P(a,c)\}\rangle$, and $\mathcal{B}_5 = \langle\{P(d,e), \neg C(d,e)\}, \{P(d,e), \neg C(d,e)\}\rangle$, determine the tree $\mathcal{T}(\mathcal{R})$ depicted in Ex. 4. Since $\mathcal{T}(\mathcal{R})$ is non-warranting and $\mathcal{R}$ is the only query supporter, we conclude $\Sigma \not\mid\!\approx P(a,b)$.

4 How Feasible Is This Non-standard DL-Reasoning Methodology?

To implement our argumentation-DL machinery two questions need to be addressed: how to construct (1) an argument supporting a query, and (2) the defeaters of a given argument. For (1) techniques upon the reasoning procedure on the DL at issue may be used. Some works on this matter are [11] (in $\mathcal{ALC}$) and [3] (in $\mathcal{EL}$). Here, we address (2) by relying on MUPS (minimal unsatisfiability-preserving sub-TBoxes) and MIPS (minimal incoherence-preserving sub-TBoxes) [12]. Such structures are defined by following an extension of the standard $\mathcal{ALC}$-tableau [2] applied to unfoldable $\mathcal{ALC}$. This algorithm is referred by the authors as *axiom pinpointing*. Next we introduce the intuitions to calculate MUPS, and afterwards extend them to propose an algorithm for recognizing defeaters of a given argument. We will rely upon unfolded $\mathcal{ALC}$ ontologies.

Unsatisfiability of a concept is detected with a labelled saturated tableau. A labelled tableau is a set of labelled branches. A labelled branch is a set of labelled formulas of the form $(a : C)^X$, where a is an individual name, C is a concept, and X is the label containing a set of axioms which leads to the inference of the formula $(a : C)$. A formula can occur with different labels on the same branch. A labelled tableau is saturated if all its branches are closed. A branch is closed if it contains a clash, *i.e.*, if there is at least one pair of formulas with contradictory atoms on the same individual. Hence, the information on which axioms are relevant for the closure (clash) of a branch is contained in the labels of contradictory formulas. For instance, a branch λ is closed if there is some pair $(a : A)^X \in \lambda$ and $(a : \neg A)^Y \in \lambda$, where A is an atomic concept. That is, axioms from X and Y lead to clashes, *i.e.*, $X \cup Y$ is unsatisfiable.

The closure of a branch is pursued by applying expansion rules which progressively unfold axioms in a lazy manner. An example of expansion rule over a branch λ is, If $(a : A)^X \in \lambda$ and $A \sqsubseteq C \in \Sigma$ then λ is replaced in the tableaux by $\lambda \cup \{(a : C)^{X \cup \{A \sqsubseteq C\}}\}$. Additional branches may be progressively included in the tableaux by following a disjunctive rule which operates over formulas like $(a : C_1 \sqcap C_2)$. (For an

account of the complete set of expansion rules, please refer to [12].) Once no more rules can be applied, a closed tableau is obtained through the set S of closed branches.

MUPS are constructed by building a labelled tableau for a branch initially containing only $(a : A)^{\emptyset}$, and by applying a *minimization function* φ which also starts in $(a : A)^{\emptyset}$. As expansion rules are applied to close the tableau, different rules also expand the minimization function. The idea is to obtain the smallest conjunction of axioms $\alpha_1 \wedge \ldots \wedge \alpha_n$, called *prime implicant*, implying φ. This is built from labels of contradictory formulae in each closed branch of the tableaux, hence $\alpha_i \in \Sigma$, for any $1 \leq i \leq n$. As φ is a minimization function every implicant of φ is also a minimization function. Finally, the prime implicant is also a minimization function. This means that A is unsatisfiable when $\alpha_1 \wedge \ldots \wedge \alpha_n$ is true, or equivalently A is unsatisfiable wrt. the set $M = \{\alpha_1, \ldots, \alpha_n\}$, which is minimal since it comes from a prime implicant (smallest conjunction of axioms implying φ). The MUPS for A wrt. Σ is $mups(A, \Sigma) = \{M \subseteq \Sigma | A \text{ is unsatisfiable in } M \text{ but } A \text{ is satisfiable in any } M' \subset M\}$.

To calculate the defeaters of a given argument $\mathcal{B}$, we first calculate $mups(A, \Sigma)$ where $A \sqsubseteq C \in \mathfrak{bd}(\mathcal{B})$. For any axiom $A' \sqsubseteq C' \in \mathfrak{bd}(\mathcal{B})$ which was not considered by the process to close the tableau (*i.e.*, not included in any label), a new MUPS $mups(A', \Sigma)$ should be obtained. (Only in the worst case a MUPS for every axiom within $\mathcal{B}$ should be constructed.) Once all the necessary MUPS are obtained, we join them into a set $mupsArg(\mathcal{B}, \Sigma) = \bigcup_{A \sqsubseteq C \in \mathfrak{bd}(\mathcal{B})} mups(A, \Sigma)$. Finally, for any $M \in mupsArg(\mathcal{B}, \Sigma)$, it holds $M \setminus \mathfrak{bd}(\mathcal{B})$ is the body of a minimal undercut of $\mathcal{B}$, conforming definitions 1, 3, and 4 (an mcu conforming Def. 5 can be easily obtained by accommodating the claim). In order to find inconsistencies beyond incoherencies, *i.e.*, to find also contrary membership assertions form the ABox, the original MUPS algorithm should consider additional expansion rules, thus turning MUPS from sub-terminologies to sub-ontologies. This assumption should not affect the following analysis.

Calculating MUPS relies on the construction of a minimization function from a tableau. Building it in a depth-first way allows to keep one single branch in memory at a time. Hence, the complexity class of the MUPS problem corresponds to that of the satisfiability checking in unfoldable $\mathcal{ALC}$, *i.e.*, PSPACE. Since the size of prime implicants may be exponential wrt. the number of axioms in the TBox, approximation methods could avoid the construction of fully saturated tableaux to reduce the size of the minimization functions. In addition, in order to render all the necessary defeaters of a given argument $\mathcal{B}$, the construction of several MUPS could be necessary. However, the unfolding process would find (in general) most of $\mathcal{B}$'s axioms.

Theorem 2. *Calculating all the defeaters of a given $\mathcal{ALC}$ argument is in PSPACE.*

5 Conclusions and Future Work

A general DL-argumentation machinery was proposed. This new ontology reasoner provides argumentation techniques to reason over inconsistent/ incoherent ontologies, and behaves as a classical ontology reasoner when considering consistent ontologies (Theorem 1). A similar $\mathcal{ALC}$-argumentation framework (without cqs support) was proposed in [13], where difficulties regarding negation of DL-axioms were addressed by specifying specialized semantics enriched with $\neg$, $\wedge$, and $\vee$, to define defeaters. From

our viewpoint, this would require to extend widely accepted $\mathcal{ALC}$-reasoning techniques with classic logic characteristics in order to build arguments in practice.

As far as classic DL-reasoning methodologies are reused to construct machineries for DL-argumentation, argumentation will provide a useful alternative to reason over inconsistent ontologies –the complexity of the argumentative reasoner will depend on that of the adopted DL-reasoner. Nonetheless, certain increase in the complexity should be presumed when working with huge dialectical trees, so establishing a relation to the "level of inconsistency" (number of contradictory axioms) of the queried ontology.

The complexity analysis of answering whether a query α is accepted by the DL-argumentation machinery requires to construct and mark the dialectical tree rooted in an α-supporter. In unfolded $\mathcal{ALC}$, this problem approaches to that of calculating MIPS from all MUPS. Moreover, since the number of defeaters for a complete dialectical tree, may grow exponentially in the number of axioms of the TBox, we believe that the construction of the tree in unfolded $\mathcal{ALC}$ would be at least in EXPTIME (as satisfiability in $\mathcal{ALC}$). A deep analysis on this matter is underway. Future work also pursues complete algorithms for $\mathcal{ALC}$ and most importantly, for efficient DL families as DL-Lite and $\mathcal{EL}$.

References

1. Baader, F.: Terminological Cycles in a Description Logic with Existential Restrictions. In: IJCAI, pp. 325–330 (2003)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): Description Logic Handbook: Theory, Implementation and Application. Cambridge University Press, Cambridge (2003)
3. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the Description Logic EL. In: Description Logics (2007)
4. Besnard, P., Hunter, A.: A Logic-based Theory of Deductive Arguments. Artif. Intell. 128 (1-2), 203–235 (2001)
5. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite family. JAR 39(3), 385–429 (2007)
6. Cecchi, L., Fillottrani, P., Simari, G.: On the complexity of DeLP through game semantics. In: NMR, pp. 386–394 (2006)
7. Dung, P.: On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning and Logic Programming and n-person Games. Artif. Intell. 77, 321–357 (1995)
8. Flouris, G., Huang, Z., Pan, J., Plexousakis, D., Wache, H.: Inconsistencies, Negations and Changes in Ontologies. In: AAAI, pp. 1295–1300 (2006)
9. García, A., Simari, G.: Defeasible Logic Programming: An Argumentative Approach. TPLP 4(1-2), 95–138 (2004)
10. Gómez, S., Chesñevar, C., Simari, G.: Reasoning with Inconsistent Ontologies through Argumentation. Applied Artificial Intelligence 24(1&2), 102–148 (2010)
11. Meyer, T., Lee, K., Booth, R., Pan, J.Z.: Finding Maximally Satisfiable Terminologies for the Description Logic ALC. In: AAAI (2006)
12. Schlobach, S., Cornet, R.: Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. In: IJCAI, pp. 355–362 (2003)
13. Zhang, X., Zhang, Z., Lin, Z.: An Argumentative Semantics for Paraconsistent Reasoning in Description Logic ALC. In: Description Logics (2009)

Planning under the Uncertainty of the Technical Analysis of Stock Markets

Augusto C.E. Baffa and Angelo E.M. Ciarlini

Departamento de Informática Aplicada
CCET - UNIRIO
Rio de Janeiro, Brazil
{augusto.baffa,angelo.ciarlini}@uniriotec.br

Abstract. The stock market can be considered a nondeterministic and partially observable domain, because investors never know all information that affects prices and the result of an investment is always uncertain. Technical Analysis methods demand only data that are easily available, i.e. the series of prices and trade volumes, and are then very useful to predict current price trends. Analysts have however to deal with the fact that the indications of these methods are uncertain, having different interpretations. In this work, we assume the hypothesis that an investment context can be modeled as a Partially Observable Markov Decision Process (POMDP) and partial observations of price trends can be provided by Technical Analysis methods. A metamodel is proposed to specify POMDP problems, embedding formal interpretations of Technical Analysis methods. Planning algorithms can then try to create investment policies to maximize profits. Due to the complexity for solving POMDPs, algorithms that generate only approximate solutions have to be applied. Nevertheless, the results obtained by an implemented prototype are promising.

Keywords: Stock Market, POMDP, Technical Analysis, Planning.

1 Introduction

The analysis of investments is a common activity in the economy. Daily, portfolio administrators use their skills to assess when and where to invest. Investors are always looking for information to improve their profits. Many players use tools of Technical Analysis [1] to support their investment decisions. Technical Analysis [1] is a discipline that seeks to predict the future direction of prices by means of the analysis of historical market data: price and trade volume. It uses statistical methods to determine the best time to buy or sell an asset. During analysis, the investor tries to identify chart patterns in price series, such as "flag", "triangles" or "head and shoulders". The analyst also uses mathematical indicators and statistical studies to further assist in her or his decision. The Technical Analysis discipline is based on Dow's Hypothesis [2] which says that asset prices reflect the market players' reaction to all relevant information. According to this theory, price discounts everything and has repetitive trends and patterns. Indicators are mathematical methods used to detect

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 110–119, 2010.

beginning and reversal of trends. There are many indicators that use different strategies to detect these events. In addition, there are several ways to interpret the indications provided by a method.

In order to help investors in their decision making process by means of the automatic application of Technical Analysis methods, considering the potential risks and benefits of a particular decision, it is necessary to formally model these methods and their interpretations. In addition, it should be noticed that previous decisions can influence the results and future decisions. In this way, we need to adopt policies that take into account the uncertainty of Technical Analysis methods and the risks and benefits of the combination of long-term decisions.

Investment analysis in the stock market can be seen as a nondeterministic problem, since we can never be sure about the results of our decisions. It is also a partially observable problem, because even experienced analysts are not able to predict all factors that can affect stock prices. We propose a metamodel for modeling investments in stocks as POMDP (Partially Observable Markov Decision Process) problems [3], so that they can be solved by planning algorithms. In Markov Processes, it is assumed that the probabilities of a transition between states depend only on the current state. In order to cope with this restriction, we model the state at each time as containing information on a time window from the recent past to the near future. Such a window is made large enough to describe the trend at each moment. Data from the near future is not known, but we assume that Technical Analysis provides partial observation of it. As demanded by a POMDP model, probabilities for transitions between states and for the correspondence between Technical Analysis indications and states have to be specified. Such probabilities are either calculated on basis of historic data or established by experienced analysts.

Planning algorithms to generate solutions for POMDPs are very time consuming and are not tractable when the number of states grow. In order to solve our POMDPs, we have adopted algorithms that generate approximate solutions and tried to describe the investment context by means of a few discrete states. In this paper, we describe the ideas about the main planning algorithm we have used, the proposed metamodel itself and the results obtained by its experimental use. Details about a tool that supports the metamodel can be found in [4].

Related works on Artificial Intelligence research have also aimed to provide support for decision making in stock markets. Neural networks, for instance, have been used either to directly suggest buying or selling assets [5, 6] or to detect Technical Analysis chart patterns [7]. Reinforcement learning has also been applied to implement portfolio selection techniques [8] and to generate decision-making rules [9]. The main difference of our approach is that it is based on creating a platform to test alternatives for modeling and planning stock investments rather than proposing a single alternative. We focus on providing flexibility to model investment contexts as POMDP problems that take into account the uncertainty of Technical Analysis, which can be used by planning algorithms to generate investment policies. Flexibility is provided to formally specify the concept of current trend and the interpretation of different Technical Analysis techniques, as well as to use different planning algorithms.

Section 2 explains concepts of POMDPs and planning algorithms for solving them. Section 3 presents the metamodel. Section 4 describes a basic model defined in accordance with our metamodel and the results obtained by this model. Section 5 contains our concluding remarks.

2 Partially Observable Markov Decision Processes

A Partially Observable Markov Decision Process (POMDP) is a framework to describe a process in which an agent performs nondeterministic actions without being able to directly observe the underlying state. The agent has to decide their actions based on some local observations and probability distributions over the states of the modeled domain. The domain should be modeled as a stochastic system with nondeterministic state transitions, caused by actions. Transitions from states by means of actions are also associated with specific probabilities [10]. More formally, a POMDP is represented by the tuple $\sum = (S, A, T, O, P, C, R)$ where:

- S is a finite set of states, A is a finite set of actions and O is a finite set of observations.
- $T_a(s' \mid s)$ is a probability distribution. For any $a \in A$, $s \in S$ and $s' \in S$, $T_a(s' \mid s)$ is the probability of reaching state *s'* after executing action *a* in *s'*. If, for any state *s'* and action *a*, $T_a(s' \mid s) \neq 0$, then $\sum_{s' \in S} T_a(s' \mid s) = 1$.
- $P_a(o \mid s)$ is a probability distribution. For any $a \in A$, $s \in S$ and $o \in O$, $P_a(o \mid s)$ is the probability of observing *o* in state *s* after executing *a*. If, for any action *a* and observation *o*, $P_a(o \mid s) \neq 0$, then $\sum_{o \in O} P_a(o \mid s) = 1$.
- $C : S \times A \rightarrow \Re$ is a function that assigns a cost to each state and action.
- $R : S \rightarrow \Re$ is a function that assigns each state to a reward.

The sets of observations and the probability distributions in *P* introduce the model of partial observability. The information on the current state is obtained by means of observations. Different states of the set *S* can result in the same observation, so that the controller might not perceive differences between states from the valid observations. .

As the current state is unknown, it is necessary to reason based on a belief, that is, a distribution probability of being at each state. Let us call the set of all possible beliefs *B*. If $b \in B$ is a belief state, the probability of being in state s is denoted by $b(s)$. The function $b_a(s)$ calculates the probability of reaching state *s*, by executing action *a* from the current belief state *b*.

$$b_a(s) = \sum_{s' \in S} T_a(s \mid s') b(s') \cdot \tag{1}$$

The function $b_a(o)$ calculates the probability of observing *o*, after having executed action *a*, from the current belief state *b*.

$$b_a(o) = \sum_{s \in S} P_a(o \mid s) b_a(s) . \tag{2}$$

So, given a belief state *b*, the execution of an action *a* and subsequent observation of *o* results in a new belief state defined by b_a^o, as described by the expression:

$$b_a^o(s) = \frac{P_a(o \mid s) b_a(s)}{b_a(o)}. \tag{3}$$

The reward for a belief state *b* when executing an action *a* is calculated by the function:

$$\rho(b,a) = \sum_{s \in S} b(s)(R(s) - C(s,a)). \tag{4}$$

A solution for a POMDP is a policy $\pi : B \rightarrow A$ that maps belief states into actions[11]. A discount factor γ is normally used to generate policies that give more importance to recent benefits and costs. An optimal policy is a policy that maximizes the possibility of future benefits, determined by the difference between rewards and costs. The POMDP planning problem is classified as an optimization problem. Its objective is the generation of a policy that determines, for each belief state b, the best action to maximize the expected benefit *E(b)* as described by the utility function (known as Bellman backup operator)[12]:

$$E(b) = \max_{a \in A} \{ \rho(b,a) + \gamma \sum_{o \in O} b_a(o) E(b_a^o) \}. \tag{5}$$

The general method for solving a POMDP problem was proposed by Smallwood and Sondik[13] and uses a Value Iteration Algorithm[14]. It calculates a complete solution for a finite horizon of *t* states using a dynamic programming procedure, based on the choice of the α function that maximizes its "scalar product" by the belief state (seeing α and *b* as |S|-dimension vectors).

$$E_t(b) = \max_{\alpha \in \Gamma_t} \{ \sum_{s \in S} \alpha(s) b(s) \}. \tag{6}$$

The complete solution of a POMDP problem generates a set Γ_t of α functions to a horizon *t* based on the set of Γ_{t-1} and considers all possible belief states. Each α function represents the action *a* that maximizes its "scalar product" by a belief state *b*. Algorithms that generate approximations are usually adopted since algorithms that return complete solutions work only with very small problems [15].

According to Pineau, Gordon, and Thrun [16], the Point-Based Value Iteration algorithm [4] is very efficient to solve POMDP problems and generates good approximate solutions. It is based on Smallwood and Sondik complete approach, but without considering all possible belief states to calculate the set Γ_t of α functions. The process starts considering only an initial belief state and progressively includes others that are reachable by means of possible transitions and observations [17].

For the current set of belief states *Bc* under consideration the set Γ_t is recursively calculated using the function $\alpha^{a,*}$ and the set of functions $\Gamma_t^{a,o}$ where $a \in A$ e $o \in O$:

$$\alpha^{a,*}(s) = R(s) - C(s,a). \tag{7}$$

$$\Gamma_t^{a,o} = \{ \gamma \sum_{s' \in S} T_a(s' \mid s) P_a(o \mid s') \alpha_i(s') \mid \forall \alpha_i \in \Gamma_{t-1} \}. \tag{8}$$

After that, we create the set Γ_t^a for the finite set of beliefs:

$$\Gamma_t^a = \{\alpha^{a,*} + \sum_{o \in O} \arg\max\{\sum_{s \in S} \alpha(s) b_i(s)\} \mid \forall b_i \in Bc\}, t > 0 \cdot \tag{9}$$

$$\Gamma_0^a = \{\alpha^{a,*}\}. \tag{10}$$

The set of solutions Γ_t contains only the α functions that maximize one of the belief states $b \in Bc$, according to the formula:

$$\Gamma_t = \{\arg\max_{\alpha_b^a \in \Gamma_t^a}\{\sum_{s \in S} \alpha_b^a(s) b(s)\} \mid \forall b \in Bc\} \cdot \tag{11}$$

The horizon t is a parameter of the algorithm and should be long enough to reach a fixed-point for Γ_t. The number of expansions of the set Bc is also a parameter of the algorithm. The optimal policy is generated by selecting the belief state and action related *to* each α element in Γ_t. A table corresponding to the belief states reached after each action and its subsequent observation is created for simulation purposes. It contains only entries for belief states in Bc. Because of the approximation, it is possible to reach a new belief state that is not in Bc. In this case, it is approximated to the nearest belief state in the policy, estimated by a root mean squared error.

3 Metamodel for Investment Contexts

We propose a metamodel to describe stock market contexts observable by means of Technical Analysis methods. The metamodel provides instructions about how states, transitions and observations are specified and probabilities of the POMDP are calculated. Each item is explained below.

States – Each state is composed of a deterministic part and a nondeterministic one. The deterministic part represents the current state of the investor's portfolio. The non-deterministic part is formed by one or more components and corresponds to the price variation, the average price, volume or quantity in a specific period. Usually, different periods are combined (e.g. a price variation from the previous week and the variation to the next week). The nondeterministic part should be specified in such a way to capture the price variation in the near future that is caused by the current trend (which is the part of the state that is not observable). The parts of the state are discrete. It is then necessary to define ranges that represent uptrends, downtrends or neutral periods.

Observations – The observations determine how a sensor or sensors interpret market information. The sensors use technical analysis indicators or patterns that map market data to discrete observations, such as estimates of high, bull market, low, bear market, flat market, oscillation and trend reversal. Sensors can combine the indications of more basic sensors to improve its results. The user can specify both the interpretation of basic sensors and the composition of complex sensors and its interpretation.

Actions and Transitions – Each action defines changes in the deterministic part of the state. Usually, the set of actions of a model contains variants of buying and selling and a dummy action *nothing*. Transitions correspond to possible changes of states, considering both deterministic and nondeterministic (stochastic) parts. The state transitions of a model are then generated by combining the possible changes in the deterministic and the nondeterministic parts.

Probabilities – The probabilities of transitions (of the nondeterministic parts of the states) and observations are generated by an analysis of the historical time series analysis. As the state at times in the past can be easily identified, the observations generated for them and the transitions that occurred can be counted in order to establish probabilities. Alternatively probabilities can be specified by experts based on their experience.

Rewards and Costs – Actions have costs and states have rewards. Rewards can be positive or negative values. States that represents profits must be assigned to positive rewards and states that represent losses to negative rewards. Costs can be assigned to actions in accordance with the fees that have to be paid to brokers. For simplicity, models can consider only rewards.

4 Basic Model and Results

A basic model was proposed in accordance with the metamodel. This model is very simple, but it already led to interesting results. More sophisticated models can, however be applied to study the importance of several variables in detail.

In the basic model, states are composed of only three variables: (a) market trend from last six days (bull, bear or flat), (b) market trend for the next six days, and (c) the investor's position (long, short or off). The first two variables are responsible for the nondeterministic part of the state. The first is observable because it is a past event and the second is not observable because it is a future event. The third variable is the deterministic part of the state and is responsible for representing the current position of the portfolio. As each variable can assume three different values, the model has 27 states. We determined that the trends will be represented by periods of six days and have assumed the following definitions in line with the price change: *bull* market for variation $\geq +2\%$, *bear* market for variation $\leq -2\%$ and *flat* market for variation in between.

Investors can be either in *short* or *long* positions towards the stock being analyzed or can be *off* the market. In a long position, the investor will have bought stock shares corresponding to a fixed amount of money M. In a short position, the investor will have borrowed and immediately sold stock shares corresponding to the amount of money M, expecting to rebuy them at a lower price before returning them to the owners. In long positions, profit is proportional the stock price variation and in short positions it is proportional to the symmetric of price variation.

The possible actions are *Buy*, *Sell*, *DoubleBuy*, *DoubleSell* and *Nothing* and are related to the deterministic component of the states. *Buy* and *Sell* actions correspond, respectively, to buying and selling an amount of money M in stock shares. *DoubleBuy*

and *DoubleSell* actions correspond to an amount *2M* and are used to directly change the investor's position from short to long and vice-versa. Action *Nothing* corresponds to keeping the investor's position.

The assignment of rewards to states is essential to specify how profitable each state is. It is highly desirable to be in a long position during a bull market and in a short position during a bear market. When the market is flat, it might be preferable to stay off the market, avoiding clearing costs and loss of other investment opportunities. We assigned smaller rewards to trends that are not so clear. States that represents "off market" are close to zero, and should be slightly positive if the market is flat and slightly negative if there is a clear trend that the investor could take advantage.

The observations are produced by monitoring the series and are based on the formal definition of each sensor. Initially, this model uses *oversold*, *overbought*, and *undefined* as observations. We selected three Technical Analysis indicators: RSI (Relative Strength Index)[18], MACD (Moving Average Convergence-Divergence)[19] and SO (Stochastic Oscillator)[20]. Each indicator has more than one interpretation and we decided to use two for each of them. Thus, as each interpretation generates a different model, and we chose 3 indicators, each period chosen to calculate probabilities generates six different versions of the basic model.

We conducted experiments based on the values of the Bovespa Index[1] (IBOV). In a first experiment, the probabilities of the POMDP model were calculated using the period from Jan. 2nd, 2000 to Dec. 30th, 2007. This period had many important events that influenced market prices, such as the dot-com bubble, Sept. 11th terrorist attacks and the Brazilian presidential election crisis in 2002. To evaluate the generated policy, we simulated its application to a period that starting on January 2nd, 2008 and ending on June 30th, 2009. It was selected because it had 3 different trends that influenced market prices. The first semester of 2008 had a flat trend. During the second semester of 2008, a bear trend occurred, influenced by the subprime crisis. The first semester of 2009 was a bullish period again, recovering previous losses.

In a second experiment (also with IBOV), the period from January 2nd, 2001 to December 30th., 2008 was used for calculating probabilities. This period includes almost all the events of the learning period used in the first experiment and the year 2008. The application of the generated policy was simulated for the period starting on January 2nd, 2009 and ending on December 30th, 2009. It covers a unique bullish trend and was interesting to evaluate policies during long-term trends.

The planning processes generated policies with about 1500 belief states. During the simulations, the observations are provided continuously to each trading day and the policy in question determines which action should be executed. Short operations were important to minimize losses and still generate profits during a crisis period. Operations where the investor takes long positions were important especially in the bull period after a crisis. The results are demonstrated in Table 1. Table 2 shows the results that would have been obtained by immediately obeying the indications of the sensors, without using a model to capture the uncertainty of Technical Analysis methods.

[1] Bovespa (Bolsa de Valores de São Paulo) is the main Brazilian stock exchange which is based in São Paulo city.

Table 1. Planner Results x Ibovespa Index

Indicators	2008 Jan-Jun	2008 Jul-Nov	2009 Dec-Jun	1st Period Total	2009 Jan-Dec
RSI1	2.06%	-10.50%	55.72%	47.27%	88.55%
RSI2	14.57%	14.39%	38.88%	67.84%	26.08%
MACD1	20.80%	0.00%	4.38%	25.18%	66.92%
MACD2	19.77%	-59.46%	28.72%	-10.97%	44.48%
SO1	10.06%	28.78%	51.83%	90.67%	9.37%
SO2	2.14%	-84.37%	55.44%	-26.79%	60.43%
IBOV	-1.35%	-36.88%	39.69%	-10.80%	70.43%

Table 2. Results for immediate obedience to indicators

Indicators	2008 Jan-Jun	2008 Jul-Nov	2009 Dec-Jun	1st Period Total	2009 Jan-Dec
RSI1	1.67%	0.00%	-5.93%	-4.26%	21.83%
RSI2	14.01%	-28.14%	25.38%	11.25%	10.78%
MACD1	1.98%	0.00%	-6.36%	-4.38%	5.54%
MACD2	14.71%	-65.53%	11.80%	-33.13%	19.36%
SO1	-7.37%	-69.06%	5.98%	-70.45%	16.43%
SO2	4.67%	-86.77%	0.00%	-82.10%	36.71%

During the first period, all indicators, except SO2 (-26.79%) and MACD2 (-10.97%) obtained a better performance than IBOV (-10.80%), although some indicators were much better than others. In general, simulations based on RSI2 (67.84%) and SO1 (90.67%) were able to make profits, even during a crisis (between July and November 2008). All indicators had better results when using the basic model POMDP in relation to the option of simply following the directions of the sensors. The second period produced good results for RSI and SO indicators, but MACD provided better results, apparently working better in well-defined trend periods. With the exception of SO1, all results were good and better than immediate obedience. As this period had essentially a long uptrend, the data were not divided into shorter periods.

5 Conclusion

This work presents an approach to the generation of investment policies in the stock market. The approach is based on modeling investment contexts as POMDP problems that take into consideration the uncertainty of Technical Analysis methods, so that planning algorithms can be applied to obtain approximate solutions. We aimed to create the means to obtain coherent policies that produce results similar to those obtained by experienced investors.

We proposed a metamodel for modeling contexts of investment in different ways, corresponding to various uses of Technical Analysis methods. The metamodel was

used to specify a basic model, which was tested with real market data. The combined use of Technical Analysis with POMDPs has produced much better results than simply following Technical Analysis indicators. The results confirmed the hypothesis that POMDPs can improve profitability because the indicators are not perfect and by considering their uncertainty we can reduce the risks. The approach proved also to be useful in comparing the concepts of technical analysis, so that an investor can choose the sensor that seems to produce better results for a specific asset. In general, results were close to those that could be obtained by a human expert investor.

In the continuation of this project, the basic model and alternative models will be applied to other stocks and indexes. Modifications of the metamodel will be investigated to study correlations between stocks and to support investment in stock options. We intend also to look into alternatives mixing our approach with machine learning techniques, either to directly produce observations or to better evaluate rewards and costs.

As the execution of the planning process for a single model takes around 12 hours on a PC, the prototype tool was implemented using a grid architecture that allowed us to test various versions of the basic model in parallel. An interesting alternative is to implement a parallel PBVI planner. It will allow the distribution of a single POMDP planning job into multiple machines. Finally, other planning algorithms can still be studied and incorporated into the tool as alternatives.

References

1. Kirkpatrick, C., Dahlquist, J.: Technical Analysis: the Complete Resource for Financial Market Technicians. First. FT Press (2006)
2. Hamilton, W.: The Stock Market Barometer: A Study of its Forecast Value Based on Charles H. Dow's Theory of the Price Movement. John Wiley & Sons, Inc., New York (1998) (reprint of 1922 edition)
3. Cassandra, A.R., Kaelbling, L.P., Littman, M.L.: Acting optimally in partially observable stochastic domains. In: Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI) Seattle, WA (1994)
4. Baffa, A.C.E.: Ambiente para Modelagem, Planejamento e Avaliação De Políticas No Mercado De Ações, Master Thesis, Universidade Federal do Estado do RJ (April 2010)
5. Tilakaratne, C.D., Morris, S.A., Mammadov, M.A., Hurst, C.P.: Predicting Stock Market Index Trading Signals Using Neural Networks. In: Proceedings of the 14th Annual Global Finance Conference, Melbourne, Australia (2007)
6. Kainijo, K.-I., Tanigawa, T.: Stock Price Pattern Recognition - A Recurrent Neural Network Approach. In: Proc. of Int. Conf. Neural Networks (1990)
7. Nevmyvaka, Y., Feng, Y., Kearns, M.: Reinforcement Learning for Optimized Trade Execution. In: Proceedings of the 23rd International Conf. on Machine Learning (2006)
8. Li, H., Dagli, C.H., Enke, D.: Short-term Stock Market Timing Prediction under Reinforcement Learning Schemes. In: IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning (2007)
9. Mabu, S., Hirasawa, K., Furuzuki, T.: Trading Rules on Stock Markets Using Genetic Networking Programming with Reinforcement Learning and Importance Index
10. Nau, D., Ghallab, M., Traverso, P.: Automated Planning: Theory & Practice. Morgan Kaufmann Publishers Inc., San Francisco (2004)

11. Kurniawati, H., Hsu, D., Lee, W.S.: SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In: Proc. Robotics: Science and Systems
12. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton (1957)
13. Smallwood, R.D., Sondik, E.J.: The Optimal Control of a Partially Observable Markov Processes over a Finite Horizon. Operations Research 21(5) (1973)
14. Cassandra, A.R.: Exact and Approximate Algorithms for Partially Observable Markov Decision Processes. Ph.D. Thesis. Brown University, Department of Computer Science, Providence, RI (1998)
15. Smith, T., Thompsom, D.R., Wettergreen, D.S.: Generating exponentially smaller POMDP models using conditionally irrelevant variable abstraction. In: Proc. Int. Conf. on Applied Planning and Scheduling (2007)
16. Pineau, J., Gordon, G., Thrun, S.: Point-based value iteration: An anytime algorithm for POMDPs. In: Proc. Int. Joint Conf. on Artificial Intelligence, Acapulco, Mexico (2003)
17. Smith, T., Simmons, R.: Heuristic search value iteration for POMDPs. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence. ACM International Conference Proceeding Series, vol. 70, pp. 520–527. AUAI Press, Arlington (2004)
18. Wilder, J.W.: New Concepts in Technical Trading Systems. Trend Research, Greensboro (1978)
19. Appel, G.: The Moving Average Convergence-Divergence Trading Method (Advanced Version). Traders Press (1985)
20. Lane, G.C.: Lane's stochastics: the ultimate oscillator. Journal of Technical Analysis (1985)

From Inconsistency to Consistency: Knowledge Base Revision by Tableaux Opening

Camilla Schwind

Laboratoire d'Informatique Fondamentale, CNRS
Université de Marseille
163 Avenue de Luminy - F-13288 Marseille Cedex09, case 901
camilla.schwind@lif.univ-mrs.fr

Abstract. We present a formal framework for theory revision that is based on a tableaux algorithm. We use a semantic tableaux system for generating the tableau of a belief base and of a revision formula. Specific revision operators can then be defined in terms of tableaux branches. Tableaux branches of a formula correspond to implicants of the formula. We define a new set of revision postulates for implicant sets and we show that they verify the AGM revision postulates. Then we consider specific distance based revision operators and in particular those based on the symmetrical difference between interpretations. We show that the minimal symmetric difference between the model sets of two implicants can be obtained by a simple binary operation on these implicants. Finally, we define three specific distance based revision operators. The representation of a formula by implicants is not unique (our approach is not based on prime implicants). We show that our implicant revision approach obtains the same revision result operated on different equivalent implicant sets for the three revision operators we propose.

1 Introduction

Knowledge base revision is an important subject in artificial intelligence and in data base systems. Considerations about the evolution of knowledge and belief are very important in many other domains such as philosophy and philosophy of science, history, law and politics. Theory revision as we will consider it in this article has been introduced in the framework of logic by Alchourron, Gärdenfors and Makinson in [2]. They considered a belief set as a deductively closed set of sentences (of a propositional logic). Revision is then an operator between a belief set and a formula that will be added to the belief set. The authors do not define specific revision operators but formulate postulates, called AGM postulates, that every reasonable revision operator should satisfy. Although the postulates have frequently been criticized they are widely used as a very usefull framework to study belief revision. The AGM postulates are very general, they specify sort of "minimal requirements" for revision operators. Katsuno and Mendelson have reformulated the postulates for belief sets which are represented by a finite set of formulas [14]. In this framework a revision is an operator between two formulas, one representing the belief base and the other representing the revision formula.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 120–132, 2010.

In this paper, we propose an approach where revision is defined as an operator on implicant sets. Intuitively, our approach is due to the following observation. Revision is the process of integrating new information into a belief base. In our framework, a belief base is represented by a set of formulas Γ of some logic. If we add new information represented by a formula μ to Γ, we consider the set $\Gamma \cup \{\mu\}$. There are two situations, either $\neg\mu \notin \Gamma$ or $\neg\mu \in \Gamma$. In the first case the AGM postulates tell us that $\Gamma * \mu$ is $\Gamma \wedge \mu$, in the second case $\Gamma \cup \{\mu\}$ is inconsistent and we must find a new set Γ' that contains μ and most of the information from Γ, that is consistent with μ. If $\Gamma \cup \{\mu\}$ is inconsistent, a tableaux theorem prover will produce a closed tableau for $\Gamma \cup \{\mu\}$, that is a tree, where every branch contains a formula and its negation. The idea underlying our approach is to "open" tableaux branches in order to obtain a consistent knowledge set. Opening a closed branch consists in suppressing the literals "responsible" of the contradiction. Since we want to keep the revision information μ we do not suppress literals coming from μ. Since we want to conserve as much information from Γ as possible, we will suppress a minimum of literals in branches of Γ just enough to obtain a consistent belief base. This minimum may be defined in different ways, yielding different revision operators.

The tableau for a formula can be seen as a set of implicants of the formula, an implicant being a finite set of literals. We define binary *revision functions* on implicant sets and we formulate implicant revision postulates that specify properties every revision function should have. We show that every implicant revision function satisfying the implicant revision postulates defines an AGM revision operator. Our new implicant revision operators verify one additional property that does not follow from the AGM postulates.

Specific revision operators are frequently defined in terms of models and most time they take into account the set of atoms in which two models differ. This means that the symmetrical difference between models must be calculated and we show that it is easily obtained by a simple operation on implicants.

We illustrate our approach by defining three specific implicant revision operators all based on the symmetric difference between models. We minimize this difference in two ways, first by set inclusion of the differences and secondly by minimizing the number of elements where the two models differ. The third operator is based on literal weights where a weight is a natural number that indicates the importance of the literal. The idea is that the higher the weight the more important is the literal in the belief base and the less a user wants to give it up. This approach offers the possibility to express preferences on literal level. It seems to us that the preference of literals is more concrete for a user than preference of models. We use the weights for revision by minimizing the sum of the weights of the symmetric difference thus giving up globally the less important information units.

This paper is organized as follows. In the next section, we recall elements and results of theory revision. In the third part, we introduce our tableaux calculus based approach and we introduce the new implicant revision postulates for which we proof a representation theorem. In the fourth part we study specific distance

based implicant revision operators. The fifth section describes some related work and section six concludes. Due to space limit, proofs could not be included in the paper.

It can happen that we use the terms knowledge set or belief base or belief set to speak of a belief base . We know that there might be subtle differences between these concepts. Nevertheless, we think that they all can be subject to implicant based revision.

2 Background

Preliminaries and Notation. We consider a propositional language over a set of propositional variables $\mathcal{P}$ (also called *atoms*). We note $\mathcal{M}$the set of all interpretations, $\mathcal{F}$ the set of all formulas, LIT the set of all literals, i.e. $LIT = \mathcal{P} \cup \{\neg a : a \in \mathcal{P}\}$. We call $\neg l$ also the opposite of l. We will identify a literal $l \in LIT$ with $\neg\neg l$, etc. as well as $\neg l$ with $\neg\neg\neg l, \ldots$, etc. For a formula $\phi \in \mathcal{F}$, $[\phi]$ is the set of models of ϕ (the set of interpretations that satisfy ϕ). We identify an interpretation with the set of atoms evaluated to true. Then $\mathcal{M} = 2^{\mathcal{P}}$[1]. Given a set of interpretations $M \subseteq \mathcal{M}$, we define $FOR(M) = \{\phi \in \mathcal{F} : M \subset [\phi]\}$ the set of formulas satisfied by all models in M. A set of interpretations M is called $\mathcal{F}$ -representable if there is a formula $\psi \in \mathcal{F}$ such that $[\psi] = M$. $FOR(\mathcal{F})$ is the set of all $\mathcal{F}$ -representable model sets. If $\mathcal{P}$ is finite then every set of models is $\mathcal{F}$ -representable, i.e. $FOR(\mathcal{F}) = 2^{\mathcal{P}}$. In this paper we will be only concerned with a language over a finite set of atoms $\mathcal{P}$. The consequence relation is noted "$\models$", i.e. $\phi \models \psi$ iff $[\phi] \subseteq [\psi]$ and we denote $Cn(\phi) = \{\psi : \phi \models \psi\}$ the set of consequences of formula ϕ. A belief base is a deductively closed set of formulas. It is easy to see that $FOR(M)$ is deductively closed, i.e. $FOR(M) = Cn(FOR(M))$. Given a finite set of formulas Γ, we denote their conjunction by $\Gamma_\wedge$.

2.1 The AGM Postulates for Belief Revision

Alchourron, Gärdenfors and Makinson proposed the well-known AGM -postulates for theory revision [2]. Here revision is defined as an operator on a *belief base* Γ and a formula μ and the result of the revision is a belief base denoted $\Gamma * \mu$. AGM postulates express what properties a revision operator should have. Revision comes with another operation, expansion, that simply adds a new information to a knowledge base regardless of whether the result is inconsistent. Expansion is noted $+$ and it holds that $\Gamma + \phi = Cn(\Gamma \cup \{\phi\})$[2]

Definition 1 (Revision operator). *Let Γ be a belief base and $\phi, \psi \in \mathcal{F}$. We call AGM revision operator every operator $*$ for which the following postulates hold*

[1] 2^M is the power set of set M.

[2] Expansion is characterized by another set of postulates and this equation is a representation theorem.

K∗1 *$\Gamma * \phi$ is a belief base.*
K∗2 $\phi \in \Gamma * \phi$
K∗3 $\Gamma * \phi \subseteq \Gamma + \phi$
K∗4 *If $\Gamma + \phi$ is satisfiable, then* $\Gamma * \phi \supseteq \Gamma + \phi$
K∗5 *If ϕ is satisfiable, then $\Gamma * \phi$ is satisfiable*
K∗6 *If* $\phi \equiv \psi$ *then* $\Gamma * \phi \equiv \Gamma * \psi$ [3]
K∗7 $\Gamma * (\phi \wedge \psi) \subseteq (\Gamma * \phi) + \psi$
K∗8 *If* $(\Gamma * \phi) + \psi$ *is satisfiable then* $(\Gamma * \phi) + \psi \subseteq \Gamma * (\phi \wedge \psi)$

Postulate K∗2 tells us that revision formula ϕ belongs to the revised base $\Gamma * \phi$. Therefore, every model of $\Gamma * \phi$ is also a model of ϕ, i.e. the set of models of $\Gamma * \phi$ is a subset of the set of models of ϕ. This means that a revision operator $*$ applied to a belief base Γ selects models from the revision formula ϕ as result of the revision operation. Thinking of revision as a selection function on model sets, we can characterize AGM postulates in terms of model sets identifying a formula (or a formula set) with the set of models satisfying it. This approach, introduced by Karl Schlechta in [20] yields a rather simpler algebraic characterization of revision operators.

In the following, $\mathcal{U}$ can be considered as a set of model sets.

Definition 2 (Revision function). *Le $\mathcal{U}$ be a set of sets.*
$sm : \mathcal{U} \times \mathcal{U} \longrightarrow \mathcal{U}$ is a revision function on $\mathcal{U}$ iff for any $M, N, L \in \mathcal{U}$

(S1) $sm(M, N) \subseteq N$
(S2) $M \cap N \subseteq sm(M, N)$
(S3) *if* $M \cap N \neq \emptyset$ *then* $sm(M, N) \subseteq M \cap N$
(S4) *if* $N \neq \emptyset$ *then* $sm(M, N) \neq \emptyset$
(S5) $sm(M, N) \cap L \subseteq sm(M, N \cap L)$
(S6) *if* $sm(M, N) \cap L \neq \emptyset$ *then* $sm(M, N \cap L) \subseteq sm(M, N) \cap L)$

The following correspondence theorems can only be formulated in this way when $\mathcal{P}$ is a finite set; If not, $FOR(M)$ is not always defined and -more important- the result of a revision is eventually not $\mathcal{F}$ -representable even when the knowledge base and the revision formula are. Given $M \subseteq \mathcal{M}$, we note $M^{\mathcal{F}} = FOR(M)_{\wedge}$ the conjunction of all formulas satisfied by all models in M.

The following correspondence results are obvious.

Theorem 1. *Let $*$ be an AGM-revision operator and M, N $\mathcal{F}$-representable model sets. Then the function sm defined by $sm(M, N) = [FOR(M) * N^{\mathcal{F}}]$ is a revision function on $2^{\mathcal{M}}$.*

Proof. S1, S2, S3, S4, S5 and S6 follow directly from K∗2, K∗3, K∗4, K∗5, K∗7 and K∗8 respectively.

Theorem 2. *Let sm be a revision function on $2^{\mathcal{M}}$. Then the operator $*$ defined by*
$\Gamma * \mu = FOR(sm([\Gamma], [\mu]))$ is an AGM-revision operator.

[3] $\phi \equiv \psi$ iff $\models \phi \leftrightarrow \psi$

Proof. K∗2, K∗3, K∗4, K∗5, K∗7 and K∗8 follow directly from S1 through S6. K∗1 and K∗6 hold because sm is defined in terms of models and $[\psi] = [Cn(\psi)]$ *for K∗1 and* $[\psi] = [\phi]$ *iff* $\psi \equiv \phi$ *for K∗6.*

As we can consider revision as an operation that integrates new information into a knowledge base by keeping as much as possible from the knowledge base state before the revision came about, it is very natural to ask the following question: Can it be the case that the revised base $\Gamma * \mu$ is just $\Gamma' \cup \{\mu\}$ for some subset Γ' of Γ?

The following consequence (FG) of the AGM postulates (shown by Gärdenfors in [12]) suggests that there is such a subset, since $\Gamma \cap (\Gamma * \phi) \subseteq \Gamma$. (SM) is the algebraic formulation.

Fact 1. (FG) $\Gamma * \phi = (\Gamma \cap (\Gamma * \phi)) + \phi$
(SM) $sm\,(M, N) = (M \cup (sm(M, N)) \cap N$

Katsuno and Mendelson characterize revision in terms of an operator on formulas [14]. The following KM postulates specialize the AGM postulates to the case of propositional logic and rephrase them in terms of finite covers for "infinite knowledge sets". Here a belief base Γ can be represented by a formula ψ such that $\Gamma = Cn(\psi)$. Let be $\psi, \mu, \psi_1, \psi_2, \mu_1, \mu_2, \nu \in \mathcal{F}$.

R1 $\psi \circ \mu$ imply μ
R2 If $\psi \wedge \mu$ is satisfiable, then $\psi \circ \mu = \psi \wedge \mu$
R3 If ψ is satisfiable, then $\psi \circ \mu$ is satisfiable
R4 If $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$ then $\psi_1 \circ \mu_1 \equiv \psi_2 \circ \mu_2$
R5 If $(\psi \circ \mu) \wedge \nu$ imply $\psi \circ (\mu \wedge \nu)$
R6 If $(\psi \circ \mu) \wedge \nu$ is satisfiable then $\psi \circ (\mu \wedge \nu)$ imply $(\psi \circ \mu) \wedge \nu$

Then the following theorem shown in [14] is straightforward:

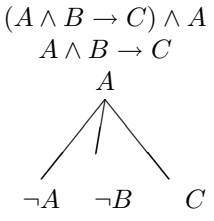
Theorem 3. *Let* Γ *be a belief base admitting a finite cover, i.e. there is* $\psi \in \mathcal{F}$ *such that* $\Gamma = Cn(\psi)$. *Then for any* $\phi, \mu \in \mathcal{F}$, $\mu \in \Gamma * \phi$ *iff* $\psi \circ \phi$ *implies* μ.

3 Semantic Tableaux and Implicant Revision

The method of semantic tableaux [21,11] has been introduced by Beth [4]. It is a syntactically oriented refutation system. A tableau rule T is of the form $\frac{\phi}{\psi_1, \ldots \psi_k}$ where $\phi, \psi_1, \ldots \psi_k \in \mathcal{F}$. In classical logic, $k \leq 2$. Tableaux rule T is applied to a formula set ϕ that belongs to a branch of a tree. If $k = 1$, ψ_1 is added to the branch. If $k = 2$, the result of the application are two new branches, one with ψ_1 added, the other with ψ_2 added. The following is a set of tableaux rules for classical propositional logic.

$$\frac{\neg\neg A}{A} \qquad \frac{A \wedge B}{A, B} \qquad \frac{\neg(A \wedge B)}{\neg A \mid \neg B}$$

A tableaux calculus can be considered as a function associating a formula tree to a set of formulas. As an example, consider the following tableau for the set of formulas $(A \wedge B \rightarrow C) \wedge A$



We consider the proof tree as a set (of branches) where each branch is a set of formulas. In the example above, the set of branches of the tree produced by the tableaux is $\{A \wedge B \rightarrow C, A, \neg A\}, \{A \wedge B \rightarrow C, A, \neg B\}, \{A \wedge B \rightarrow C, A, C\}$.

A tableau for formula ϕ can be associated with a disjunctive normal form (DNF) of ϕ, namely, each branch corresponds to the conjunction of the formulas on it (and this conjunction is equivalent to the conjunction of the literals on the branch). And formula ϕ is equivalent to the disjunction of the conjunction of its branches.

We conceive a tableaux system as a procedure TP that applies to a formula set and produces a set of sets of literals. TP is defined recursively as follows[4]:

$TP(\Gamma) = \{\Gamma\}$ if Γ is a set of literals.
$TP(\Gamma \cup \{\neg\neg\phi\}) = TP(\Gamma \cup \{\phi\})$.
$TP(\Gamma \cup \{\phi \wedge \psi\}) = TP(\Gamma \cup \{\phi\} \cup \{\psi\})$.
$TP(\Gamma \cup \{\phi \vee \psi\}) = TP(\Gamma \cup \{\phi\}) \cup TP(\Gamma \cup \{\psi\})$.

For the above formula, we obtain $TP(\{A \wedge B \rightarrow C) \wedge A\}) = TP(\{A \wedge B \rightarrow C), A\}) = TP(\{\neg A, A\}) \cup TP(\{\neg B, A\}) \cup TP(\{C, A\}) = \{\{\neg A, A\}, \{\neg B, A\}, \{C, A\}\}$

A branch is *closed* when it contains a literal l and its negation $\neg l$ and it is called *open* otherwise. A set of branches is closed when all of its elements are closed and it is open when at least one of its elements is open.

For the example above, we obtain the DNF $(A \wedge \neg B) \vee (A \wedge C)$ (taking into account that a closed branch is equivalent to $\bot$).

The fundamental property of tableaux for theorem proving is the following completeness theorem:

Theorem 1 *ϕ is a theorem, i.e. $\models \phi$ iff $TP(\{\neg\phi\})$ is closed. (Proof in [21]).*

Corollary 1. *ϕ is satisfiable, iff $TP(\{\phi\})$ is open.*

We consider TP not only as a theorem prover (or consistency checker) for propositional formulas but also as an operator that has useful properties for formulas and formula sets. We will use the following properties of TP.

[4] This formulation takes into account that for propositional logic, every rule is applied at most once to each formula.

Property 1. *Let* $\phi, \psi \in \mathcal{F}$ *,*

1. $TP(\{\phi \vee \psi\}) = TP(\{\phi\}) \cup TP(\{\psi\})$
2. $TP(\{\phi \wedge \psi\}) = \{X \cup Y : X \in TP(\{\phi\})$ *and* $Y \in TP(\{\psi\})\}$
3. *for all* $t \in TP(\phi)$, $t_\wedge \rightarrow \phi$

We will use $TS(\{\phi\})$ for $TP(\{\phi\})$with closed branches suppressed, i. e. $TS(\{\phi\}) = TP(\{\phi\}) \setminus \{t : t \in TP(\{\phi\})$ and t closed$\}$.

We introduce the following operator $\otimes$ on sets of literal sets. Let S, T be sets of literal sets. Then $S \otimes T = \{s \cup t : s \in S$ and $t \in T\}$. Then a conjunction can be considered as an operation on sets of sets (of literals): $TP(\{\phi \wedge \psi\}) = TP(\{\phi\}) \otimes TP(\{\psi\})$.

The tableau of a formula ϕ converts ϕ to a disjunctive normal form (DNF), which is equivalent to a disjunction of conjunctions of literals. Each of these conjuncts is an *implicant* of the original formula, i. e. if $\phi \equiv \phi_1 \vee \ldots \phi_n$ then $\phi_i \rightarrow \phi$ for every i (property 1, 3). It should be noted that equivalent formulas might have different (equivalent) DNF.

Subsequently, we will call *implicant* a finite set of literals. Let us note $\mathcal{T}$ the set of all implicants. A tableaux prover then produces a finite set of implicants for a formula. We can think of an implicant as being a formula, namely the conjunction of its literals and of a set of implicants as of a disjunction, namely the disjunction of the conjunction of its elements. Let t be an implicant and T a set of implicants. Then, by abuse of notation, we write $[t]$ for the set of models of the implicant formula t and $[T]$ for the set of models of the disjunction of the elements of T. An implicant set T is closed iff $[T] = \emptyset$. The following holds for $\otimes$:

Fact 2. *Let* $S, T \subseteq \mathcal{T}$ *. Then* $[S \otimes T] = [S] \cap [T]$.

In the finite case, every interpretation m can be represented by an implicant $m^I = m \cup \{\neg a : a \notin m\}$[5]. For a set of interpretations M, we set $M^I = \{m^I : m \in M\}$. Then we have $[m^I] = \{m\}$ and $[M^I] = M$.

As we have seen, belief revision is characterized by postulates that specify properties a revision operator should have. Here we will give a characterisation of revision operators in terms of implicants. The postulates we define satisfy the AGM postulates but they are stronger. They satisfy additional properties that do not follow from the AGM postulates, namely the following:

- if $\mu \in ((\Gamma * (\phi \vee \psi)) * \chi$, then $\mu \in (\Gamma * \phi) * \chi$ or $\mu \in (\Gamma * \psi) * \chi$;
- if $\mu \in (\Gamma * \phi) * \chi$ and $\mu \in (\Gamma * \psi) * \chi$ then $\mu \in (\Gamma * (\phi \vee \psi) * \chi$.

These properties are also true for distance based revision and do not follow from the AGM postulates [16].

An implicant revision operator is defined on pairs of implicant sets and its result is a new set of implicants.

Definition 3 (Implicant Revision Function). *Let be* $S, T, R \in 2^{\mathcal{T}}$ *, where* $\mathcal{T}$ *is the set of all implicants. An implicant revision operator over* $2^{\mathcal{T}}$ *is a binary function* $st : 2^{\mathcal{T}} \times 2^{\mathcal{T}} \longrightarrow 2^{\mathcal{T}}$ *for which the following postulates hold:*

[5] Remember that we identify an interpretation with the set of atoms evaluated to true.

(ST1) *For all* $v \in st(S,T)$ *there is* $t_1 \in S$, $t_2 \in T$ *and* $s \subseteq t_1$ *such that* $v = t_2 \cup (t_1 \setminus s)$.
(ST2) *If* $S \otimes T$ *is not closed then* $st(S,T) = S \otimes T$.
(ST3) *If* T *is not closed then* $st(S,T)$ *is not closed.*
(ST4) *If* $st(S,T) \otimes R$ *is not closed then* $st(S,T) \otimes R = st(S, T \otimes R)$.
(ST5) *If* $[S_1] = [S_2]$ *and* $[T_1] = [T_2]$ *then* $[st(S_1,T_1)] = [st(S_2,T_2)]$.

We will show that every implicant revision function defines a revision function.

Theorem 4. *Let st be an implicant revision function over* $2^{\mathcal{T}}$ *. Then the function* $sm : 2^{\mathcal{M}} \times 2^{\mathcal{M}} \longrightarrow 2^{\mathcal{M}}$ *defined by*

$$sm(M,N) = [st(M^I, N^I)] \tag{1}$$

is a revision function.

Postulate (ST1) tells us the following. When we revise a belief base seen as a set of implicants S by a new information, given as a set of implicants T then we choose a subset among the implicants of T and eventually augment each implicant choosen by subsets of implicants in S "compatible" (not contradictory) with that implicant. In other words, we try to "keep" as many literals occurring in implicants of the original belief base as possible.

4 Specific Revision Functions Based on Distances

Implicant revision postulates suggest to consider specific revision operations that are defined as operations on implicants. They indicate how to calculate effectively results of revision operations. Given a belief base with finite cover ϕ and a revision information μ we first produce the DNF of ϕ and μ and we obtain implicant sets S and T. Then we apply operations on the implicants of ϕ and μ in order to obtain a revised base as a new implicant set. According to postulate (ST1), every of the resulting implicants contains an implicant of μ augmented by a subset of some implicant of ϕ. Here we will concentrate on a specific class of implicant revision functions based on distance functions. Various distance based approaches to belief revision (and to update) have been defined by means of the symmetric difference between valuations, e.g. [6,23,7,19,22].

The tableaux prover will be used to calculate specific revision functions, namely those based on symmetric differences between models. Our algorithm proceeds as follows: Given a belief base ϕ and a revision formula ψ, if $\phi \wedge \psi$ is satisfiable then $\phi * \psi = \phi \wedge \psi$. In other words, if $TS(\{\phi\}) \otimes TS(\{\psi\})$ is open then $\phi * \psi = \phi \wedge \psi$. If not, we have $TS(\{\phi\}) \otimes TS(\{\psi\})$ is closed. Let be $v \in TS(\{\phi\}) \otimes TS(\{\psi\})$. Then $\exists s \in TS(\{\phi\}), t \in TS(\{\psi\})$, and $v = s \cup t$, by property 1. v is closed, s and t are open and we define the *opening* of v in s by $(s \setminus \bar{t}\,) \cup t$. This means that we have suppressed in s the literals in conflict with literals in t, i. e. the literals $s \cap \bar{t}$. Then we will define the opening of the tableaux $TS(\{\phi\}) \otimes TS(\{\psi\})$ by the set of branches "minimally opened", i.e. we keep only branches within which a minimal set (or number) of literals have been suppressed. In order to determine these minimal branches we need the following definitions.

Notation. Given a literal l, we note $|l|$ the atom of l, i.e. $|a| = |\neg a| = a$ for $a \in \mathcal{P}$. For an implicant t, we note $|t| = \{|l| : l \in t\}$. If M is a set and $\leq \subseteq M \times M$ is a pre-order on M, then $Min_<(M) = \{m \mid m \in M \text{ and } \neg\exists n \in M \text{ and } n \leq m\}$.

We first define the symmetric difference between two implicants s and t by $|s \cap \bar{t}\,|$. It turns out that $|s \cap \bar{t}\,|$ is exactly the minimal symmetric difference between the models of s and t (lemma 1). It is easy to see that $|s \cap \bar{t}\,|$ is $\subseteq$-minimal within the symmetric differences between the models of s and t.

Lemma 1. *Let be $s, t \in \mathcal{T}$. Then*
(i)$|s \cap \bar{t}\,| = |t \cap \bar{s}\,|$ and
(ii)$\{|s \cap \bar{t}\,|\} = Min_\subseteq(\{m \div n : m \in [s] \text{ and } n \in [t]\})$

Example 1. Let be $s = (a, \neg b, c)$ and $t = (\neg a, b, d)$; $[s] = \{\{a, c\}, \{a, c, d\}\}$ and $[t] = \{\{b, d\}, \{b, c, d\}\}$. Then $s \cap \bar{t} = \{a, \neg b\}$, $t \cap \bar{s} = \{\neg a, b\}$, $|s \cap \bar{t}\,| = |t \cap \bar{s}\,| = \{a, b\}$ and $Min_\subseteq(\{m \div n : m \in [s] \text{ and } n \in [t]\}) = \{a, b\}$

Any formula ϕ is the disjunction of a finite set of implicants. This defines the set of its models as a finite union of model sets (not necessarily disjoint), namely the models of its implicants. In order to obtain the $\subseteq$-minimal symmetric differences of all the models of ϕ it is sufficient to obtain the minimal sets among the $|s \cap \bar{t}\,|$ for all $s \in S$ and $t \in T$.

We will need the following lemma to show that this minimization obtains the global minimal symmetric differences for all models of the formula.

Lemma 2. *Let M be a set that is a finite union of sets (not necessarily disjoint), $M = A_1 \cup \cdots \cup A_k$ and let $\leq$ be a pre-order on M for which the limit assumption holds (i.e. the infimum of $\mathcal{M}$is a minimum). Then*

$$Min_\leq(M) = Min_\leq(\bigcup_{i=1}^{k} Min_\leq A_i)$$

The following theorem shows that the set of minimal elements of the symmetric differences between two model sets can be obtained by minimizing all sets $|s \cap \bar{t}\,|$.

Theorem 5. *Let $S, T \subseteq \mathcal{T}$ and $MinI = Min_\subseteq(\{|s \cap \bar{t}\,| : s \in S \text{ and } t \in T\})$. Then $MinI = Min_\subseteq(m \div n : m \in [S] \text{ and } n \in [T])$.*
Proof due to lemma 1 and lemma 2.

Theorem 5 shows that we obtain all minimal symmetric difference sets between the models of a formula ϕ and a formula ψ as the $\subseteq$ −smallest sets $|s \cap \bar{t}\,|$ for implicants s of ϕ and t of ψ.

Corollary 2. *Let be $S_1, S_2, T_1, T_2 \subseteq \mathcal{T}$ and $[S_1] = [S_2]$ and $[T_1] = [T_2]$. Then $Min_\subseteq(\{|s \cap \bar{t}\,| : s \in S_1 \text{ and } t \in T_1\}) = Min_\subseteq(\{|s \cap \bar{t}\,| : s \in S_2 \text{ and } t \in T_2\})$*

We define the following revision function based on the symmetric difference of models.
$st\,1(S, T) = \{t \cup (s \setminus \bar{t}\,) : t \in T, s \in S \text{ and } |s \cap \bar{t}\,| \in MinI\}$.

This means that $st\,1(S, T)$ is defined as the set of minimal openings of $S \otimes T$ in S.

Example 2. Let be $S = \{\{a, b, c\}, \{d, e, f\}\}$ and $T = \{\{\neg a, \neg b, c, d, \neg e\}\}$. Then $|\{a, b, c\} \cap \{a, b, \neg c, \neg d, e\}| = \{a, b\}$ and $|\{d, e, f\} \cap \{a, b, \neg c, \neg d, e\}| = \{e\}$. Then $st\,1(S, T) = \{\{\neg a, \neg b, c, d, \neg e, f\}, \{\neg a, \neg b, c, d, \neg e\}\}$

The following two revision functions are also based on model distance, but use the order of the cardinalities of the symmetric model differences. The first one, $st\,2$ is the well well-known distance based revision function introduced by Dalal in [7]. It uses the Hamming distance between models.

Let be $S, T \in \mathcal{T}$ and $MinH = Min_{\leq}(\{card(s \cap \bar{t}\,) : s \in S, T \in T\})$. $st\,2(S, T) = \{t \cup (s \setminus \bar{t}\,) : t \in T, s \in S \text{ and } card(s \cap \bar{t}\,) = MinH\}$

st2 is precisely the revision operation introduced by Dalal.

The revision function we propose takes into account that for a user the different entities may have different importance. Using this function presupposes that an agent associates weights to literals that measure the "importance" of the literal in the belief base. The weight of a set of literals is simply the sum of its elements. The revision operator that uses these weights is then defined in the following way. Let be w a function from LIT to $\mathcal{N}$ (set of natural numbers). Then we extend w to a function on implicants by $w(t) = \sum_{l \in t} x(l)$. Let be $MinW = Min_{\leq}(\{w(s \cap \bar{t}\,) : s \in S, t \in T\})$ The corresponding revision function is then:
$st\,3(S, T) = \{t \cup (s \setminus \bar{t}\,) : t \in T, s \in S \text{ and } w(s \cap \bar{t}\,) = MinW\}$

Theorem 6. *$st1, st2$, and $st3$ are implicant revision functions.*

4.1 Complexity

Eiter and Gottlob have shown that the problem of deciding whether a formula belongs to a knowledge base Γ after revision with μ, $\Gamma * \mu$, resides on the second level of the polynomial hierarchy [10]. Here we give the number of steps for calculating the set $st\,(S, T)$, where S and T are the set of implicants of formulas ϕ and μ. Let be Γ a formula representing a belief base and μ the new infomation. Let be n the lenght of the formula $\Gamma \wedge \mu$. Then, the number of conjuncts of a DNF of $\Gamma \wedge \mu$ is $O(2^n)$. To determine the $\subseteq$-smallest element within the subsets $s \cap \bar{t}$ we have to compare all branches pairwise, that gives polynomial time algorithm (in an exponential number of branches).

5 Related Work

Specific implicant revision operators defined for knowledge bases in specific syntactical forms specific have been proposed by several authors [17,18,5]. Perrussel et al. define a specific prime implicant revision method. Their system presupposes a knowledge base in clausal form and they propose to weighten a literal by counting the number of its occurrences within all clauses: the number of occurrences indicates the importance of the literal. The weight of an implicant is then the sum of the weights of the literals it contains. Their algorithm is based on

prime implicants; from a belief base first the set of all its prime implicants must be calculated. But the base resulting from a revision is in general not more in form of prime implicants and those have to be recalculated after every revision step, in the case of multiple revisions. They have studied the performances of their system with many benchwork examples. Bienvenu et al. [5] propose a prime implicate-based revision operator that is a full meet operator. Their algorithm is based on the set of all prime implicates of a formula K, $\Pi(K)$. The revision operator is defined by $K *_{\Pi} \phi = \phi \wedge \bigvee(\Pi(K) \perp \neg\phi)$ where $K \perp \phi$ is the set of maximal subsets of K consistent with $\neg\phi$. Again, one problem is that $\Pi(K)$ has to be recalculated after every revision step since the revised belief base is not more the conjunction of all its prime implicates. Our approach is more general (and simpler), since we do not need a belief base in normal form and we do not need to calculate the prime implicants (or prime implicates) of the belief base and the revision formula. The tableaux prover we use produces just DNF of these formulas. While the number of branches is exponential in the length of the formula, it is always less important than the set of all prime implicants. And the algorithm to obtain prime implicants is more expensive than a tableaux algorithm. But the important difference is that in all these approaches the set of all prime implicants/implicates has to be recalculated after every revision step. In our approach the DNF is calculated once for the two formulas and the revision result is still in DNF. Therefore subsequent revision operations can directly operate on the revised base and need only to produce the DNF of the revision formula. Our approach applies to an important class of revision operators, namely those based on symmetric differences of models. But there are certainly revision operators (and also distance based revision operators) that cannot be treated by our approach without a transformation of the belief base and the revision formula to prime implicants.

6 Conclusion and Further Work

In this paper we proposed to calculate belief base revision operators by means of analytic tableaux. For that purpose we have redefined the AGM postulates in terms of implicants and we have proposed a new postulate that is more specific than the AGM postulates. We have applied our approach to two specific families of revision operators. The first defines the distance between interpretations by means of their symmetric difference. The second is based on literal weights.

Here we have only treated one type of theory change operators, namely revision. Other operators are update [13], erasure or contraction [12] but also merging [15]. Our method can also be applied to these types of operators.

Another issue is iterated revision. Darwiche and Pearl in [8] have formulated postulates which, together with the AGM postulates for revision specify properties that a knowledge base must have after subsequent revision steps. It is possible to formulate postulates for iterated revision in terms of implicants and to calculate specific iterated revision operators by means of tableaux.

Another work will be to find tableaux rules that give directly a decision procedure for the problem wether a revised knowledge base $\Gamma * \phi$ contains some formula ψ. Delgrande, Jin and Pelletier in [9] have investigated rules for calculating updates that break up the update formula. We think of tableaux rules that break up the belief base formula according to the new information. Our new implicant revision postulate ST1 suggests that the revised belief base is composed of implicants that contain implicants of the revision formula and parts of implicants of the belief base. Thus, a decision procedure for the problem whether a formula is entailed by a revised belief base must modify the original belief base according to the contradictions with elements of the revision formulas.

References

1. Proceedings of the 7th National Conference on Artificial Intelligence (1988)
2. Alchourron, C.E., Gärdenfors, V., Makinson, D.: On the logic of theory change: Partial meet contradictions and revision functions. Journal of Symbolic Logic 50(2), 510–530 (1985)
3. Allen, J.A., Fikes, R., Sandewall, E. (eds.): Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (1991)
4. Beth, E.W.: The Foundations of Mathematics. North Holland, Amsterdam (1959)
5. Bienvenu, M., Herzig, A., Qi, G.: Prime implicate-based belief revision operators. In: Ghallab, M., Spyropoulos, C.D., Fakotakis, N., Avouris, N.M. (eds.) ECAI. Frontiers in Artificial Intelligence and Applications, vol. 178, pp. 741–742. IOS Press, Amsterdam (2008)
6. Bordiga, A.: Language features for flexible handling of exceptions in information systems. ACM Transactions on Database Systems 10, 563–603 (1985)
7. Dalal, M.: Investigations into a theory of knowledge base revision. In: AAAI 1988 [1], pp. 475–479 (1988)
8. Darwiche, A., Pearl, J.: On the logic of iterated belief revision. Artificial Intelligence 89(1-2), 1–29 (1997)
9. Delgrande, J.P., Jin, Y., Pelletier, F.J.: Compositional belief update. J. Artif. Intell. Res. (JAIR) 32, 757–791 (2008)
10. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates and counterfacts. Artificial Intelligence 57, 227–270 (1992)
11. Fitting, M.: First Order Logic and Automated Theorem Proving. Texts and Monographs in Computer Science. Springer, Heidelberg (1990)
12. Gärdenfors: Knowledge in Flux: Modeling the Dynamics of Epistemic States. Bradford Books. MIT Press, Cambridge (1988)
13. Katsuno, H., Mendelzon, A.O.: On the difference between updating a knowledge base and revising it. In: Allen et al. [3], pp. 387–394
14. Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. Artificial Intelligence 52, 263–294 (1991)
15. Konieczny, S., Pérez, R.P.: On the logic of merging. In: KR 1998, pp. 488–498 (1998)
16. Lehmann, D.J., Magidor, M., Schlechta, K.: Distance semantics for belief revision. J. Symb. Log. 66(1), 295–317 (2001)
17. Perrussel, L., Marchi, J., Bittencourt, G.: Une nouvelle mesure de changement minimal. In: RFIA (ed.) RFIA 2006 Proceedings Électroniques (2006)

18. Perrussel, L., Marchi, J., Bittencourt, G.: Prime implicants and belief update. In: Chad Lane, H., Guesgen, H.W. (eds.) FLAIRS Conference. AAAI Press, Menlo Park (2009)
19. Satoh, K.: Nonmonotonic reasoning by minimal belief revision. In: Proceedings of the International Conference on Fifth Generation Computer Systems, Tokyo, pp. 455–462 (1988)
20. Schlechta, K.: Coherent Systems. Studies in Logic and Practical Reasoning, vol. (2). Elsevier, Amsterdam (2004)
21. Smullyan, R.M.: First Order Logic. Springer, Heidelberg (1968)
22. Williams, M.-A.: Transmutations of knowledge systems. In: Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning, pp. 619–629 (1994)
23. Winslett, M.: Reasoning about action with a possible models approach. In: AAAI 1988 [1], pp. 89–93 (1988)

Nondeterministic Planning for Generating Interactive Plots

Fabio A. Guilherme da Silva, Angelo E.M. Ciarlini, and Sean W.M. Siqueira

UNIRIO, Department of Applied Informatics, Brazil
{fabio.silva,angelo.ciarlini,sean}@uniriotec.br

Abstract. Interactive storytelling is a new kind of digital entertainment in which users interact with a story being told. In this context, automated planning algorithms are interesting alternatives to create plots that are coherent with the intended genre. Diversity of stories and opportunities for interaction are however other key requirements to be considered by such applications. It is also important to generate stories incorporating failed attempts to achieve goals, since this is essential to create dramatic tension. In this paper, we describe a planner that takes these issues into account by means of planning with nondeterministic events, that is, events with more than one possible different outcome. The implemented planner has been incorporated to the new version of the interactive storytelling system Logtell.

Keywords: Interactive Storytelling, Planning, HTN, Nondeterminism.

1 Introduction

Planning is the subarea of Artificial Intelligence that studies the abstract and explicit deliberation process of choosing and organising actions, by anticipating their expected outcomes, in order to achieve objectives [1]. Due to the wide range of possible practical applications, planning is a field of fundamental importance in Artificial Intelligence research. The vast majority of research involving planning concerns the so-called *classical planning*. This approach requires complete knowledge about an environment that should be (modelled as) deterministic, static and should contain a finite number of states. In this approach, goals correspond to the achievement of states that satisfy predefined conditions and the time is implicit (i.e. actions change the world instantaneously). Moreover, the generated plans are expected to be linearly ordered finite sequences of actions. The adoption of these restrictions leads to a simplification that is very different from the real world, but it is still useful for some domains. It also allowed the study of domain-independent planning methods, which can be adapted to less restricted models.

One of the areas in which planning has been successfully applied is the development of systems for interactive storytelling. Interactive storytelling is a new form of digital entertainment that combines techniques and tools for creating, viewing and controlling interactive stories by electronic means [2].

The replay value of interactive storytelling systems is strongly associated with the diversity of the stories it can generate. Thus, it would be desirable to generate plots

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 133–143, 2010.

with alternatives that can possibly result in more than one alternative story. The use of planning techniques in interactive storytelling allows sequences of events to be automatically generated, providing diversity and coherence for the stories and then enhancing the user's experience. Even though different deterministic plans can be either previously generated or generated at real-time for different situations, the flexibility to obtain different coherent stories tend to be more restricted if events have only deterministic outcomes. In addition, to introduce dramatic tension in the story, it is important to incorporate events in which characters try to achieve goals without succeeding. A planning algorithm relaxing the assumption of determinism seems then to be more suitable to increase the opportunities for user interaction and the diversity of stories. The main problem is that the complexity of such an algorithm tends to be higher due to the fact that all possible contingencies resulting from an event have to be considered.

In this paper, we describe a new model for the use of nondeterministic planning to generate plots with nondeterministic events, so that the same plot can generate different stories depending on the effects assigned to those events in the dramatisation process. The process also explicitly deals with weaker goals in which characters try to achieve a certain situation, but can either succeed or fail. The model has been incorporated to Logtell [3; 4], a logic-based interactive storytelling system. To assure that the nondeterminism doesn't affect the system's overall performance, specific techniques were introduced to speed up plan generation. The main technique was the introduction of a Hierarchical Task Network (HTN) planning phase, which uses knowledge about the domain of the stories to cope with the creation of nondeterministic plans.

Section 2 gives an overview on nondeterministic and HTN planning [5]. Section 3 examines the use of planning in storytelling systems. Section 4 presents the proposed model. Section 5 describes how this model was implemented and its evaluation. Section 6 contains concluding remarks.

2 Nondeterministic Planning

Although classical planning has been helpful in some areas and applications, there are practical situations that do not offer conditions that are consistent with all its limitations. An alternative for classical planners are the partial-order planners that generate plans in which events are only partially ordered, providing more flexibility for the execution. To allow the handling of most practical problems, we need to relax other assumptions.

One of the relaxations that gives us a more realistic approach is the acceptance of nondeterminism, i.e. an action can have more than one alternative effect. In this case, a plan can result in many execution paths, requiring efficient ways to analyse all possible outcomes and the generation of plans that have conditional behaviours and trial-and-error strategies. In this context, plans usually correspond to *policies*, establishing what action should be performed when a particular state of the world is reached.

Amongst the types of nondeterministic planning, two can be highlighted: planning based on Markov Decision Processes (MDP) [6] and planning based on model checking. The first type requires probabilities associated with the possible effects of each

action and the resolution of an optimisation problem by using these values. Besides demanding the specification of probabilities that might not be known, the solution of MDPs tends to be very time consuming to be applied at real-time by an interactive storytelling system. In the second type, planning occurs under the assumption of Knightian uncertainty [7], i.e. without probabilities associated with the effects of an action. The main example of planning as model checking is the Model Based Planner (MBP) [8], which is able to handle different kinds of problems in nondeterministic domains. MBP is also able to deal with extended goals, i.e. conditions that must hold during the plan's execution path, which are expressed by the means of temporal logic formulae.

Another way to facilitate the resolution of practical problems is the use of specific knowledge about the domain of the problem to be solved. This is the case of Hierarchical Task Network (HTN) planning, in which a sequence of actions is established on the basis of *tasks* to be performed, rather than *goals* to be achieved. Informally, a task is what should be done to solve a particular problem for HTN planning. The different ways how these tasks can be accomplished are described by *methods*, and each of them establishes the decomposition of a task into a *task network* – a set of smaller tasks, or *subtasks*, totally or partially ordered. A task can have more than one possible way to be performed and, if so, the methods' definitions are used by the planner to choose which one is appropriated. By recursive application of methods to *nonprimitive tasks* (those decomposable into other tasks), we end up reaching *primitive tasks* (planning operators directly related to the problem domain), which are then bound to actions to be incorporated into the plan. Tasks and methods descriptions are based on the knowledge about the domain, which produces plans far more efficient than in classic planning. This is one of the reasons why HTN is one of the most popular planning techniques in practical applications. An example of HTN planning implementation is SHOP2, a sound and complete planner that allows the subtasks of each method to be partially ordered [9].

An interesting idea to deal with the inefficiency introduced by the nondeterminism is the use of HTN techniques. In [10] a technique is described to be used with deterministic planners, adapting them to work with nondeterminism. This technique was applied to SHOP2, generating a nondeterministic HTN version called ND-SHOP2. Experimental comparisons between ND-SHOP2 and MBP showed that, in some cases, while the CPU time of the latter grows exponentially in the size of the problem, the CPU time of the first grows polynomially.

Another work using a mixed approach is the planning algorithm Yoyo [11] which, as ND-SHOP2 does, uses an HTN-based mechanism to restrict its search and, just like the MBP planner does, makes use of model checking to deal with nondeterminism. Experimentally compared to both, Yoyo was always much faster than at least one of the two and almost always faster than both.

Both Yoyo and ND-SHOP2 plan for the achievement of a state that satisfies a predefined goal. Besides the specification of the goal, they receive as input a network of (nonprimitive) tasks to be performed in order to achieve the goal. The decomposition of such tasks is then used to guide the search for a plan. Differently from MBP, these algorithms do not allow the generation of plans with extended goals, which are interesting for interactive storytelling to create plans that satisfy conditions in intermediate states and to incorporate the idea of planning *attempts to achieve goals*. On the other

hand, MBP does not allow the specification that a certain action should occur and has also efficiency problems when executed at real-time by an interactive storytelling system. The approach presented in this paper tries to combine nondeterminism, extended goals and efficiency in accordance with the needs of an interactive storytelling system that generates and dramatises plots at real-time.

3 Planning in Interactive Storytelling

In interactive storytelling systems, two important aspects are usually in conflict. On the one hand, there is the control afforded to the user, i.e. the autonomy he has to interact with the system; on the other hand, we have the coherence of the narrative, allowing the user to understand the relationships between the events in the story [12]. Strategies for ensuring coherence usually reach their goal by depriving to the user a higher degree of freedom in how he or she can change the course of the narrative, whereas strategies aimed most at user control tend to generate threats to the coherence of the narrative. This conflict shows the difficulty of giving the user a satisfactory degree of control without affecting the coherence of the narrative, and brings forth two main approaches to the development of interactive storytelling. One of them is character-based, being inspired in games [13; 14; 15]. In this case, autonomous agents, each with its own goals, interact with each other, the environment and the user. The story itself is the result of these interactions; it causes the user to have a large degree of freedom, but it is difficult to maintain the coherence of the narrative.

The other approach, plot-based, is influenced by Literature [16] and Filmmaking and uses a strong control over the flow of the story being narrated, preventing the user from deviating much from the original intention of the author, who defines a structure for the narrative with well-defined points; the effect of user interaction being usually limited to the way the story reaches these points [17; 18]. To combine control with consistency, some solutions integrate both plot-based and character-based characteristics. Façade [19] and Erasmatron [20] are some examples of this hybrid approach.

The main benefit of planning in interactive storytelling is the possibility of automatically generating sequences of events, providing a greater degree of autonomy and dynamism. In [13], hierarchical task network (HTN) planning is used to control the way characters achieve their goals subject to user intervention. HTN planning favors efficiency at the cost of less generality, and requires the previous construction of a task hierarchy and methods to perform each task. In Façade**,** a reactive planning language is used to emulate the personality of believable agents. In Mimesis [15], a planner combining HTN and partial-order planning is used to create a storyline beforehand. Mediation techniques are used at run time to guarantee coherence, including the adoption of alternative story lines or, alternatively, interventions for forcing the failure of users' actions.

Another example of planning applied to interactive storytelling is Logtell [3; 4], an interactive storytelling system that is able to create plots based on a logical specification of a particular literary genre and the initial situation of the story. Logtell has also plot-based and character-based features and extensively uses planning to generate alternative stories at real-time. The logical specification defines the events that can occur and rules establishing goals to be pursued by characters when certain situations occur.

The Interactive Plot Generator (IPG) is the module of the system that is responsible for generating the plot. It uses a partial-order planner to create plots as a sequence of chapters. Each chapter corresponds to a cycle in which user interference is incorporated, goals are inferred and planning is used to achieve the goals. When IPG concludes that all goals have been achieved, the chapter is ready to be presented to the user. During the presentation of a chapter, the user can ask IPG to generate a different alternative for that chapter or for one previously presented. Strong interventions corresponding to the insertion of events and situations in the next chapter can also be specified. Another form of user interference is by forcing the occurrence of events and situations in the upcoming chapters. Currently, Logtell's interaction is focused in the generation of plots and, as long as one chapter is generated and incorporated into the story, the user does not intervene directly in the dramatisation of the current chapter. Interactions affect only the future chapters. In this way, the system is able to present a 3D dramatisation of the current chapter while the next chapter is being planned. Special strategies are adopted to avoid interruption in the presentation flow between chapters.

In order to allow users to interfere on the current event of a story that is being dramatised, mediation, as in Mimesis, forces the dramatisation to return to the basic line of the story or to adopt an alternative previously generated. In this case, however, all alternatives have usually to be planned beforehand and diversity tends to be limited. A more flexible approach could generalise the process of chapter creation of Logtell to plan nondeterministic chapters at real-time. In this case, however, the planner should be able to create alternatives for all continuations of the current chapter while it is being presented. Another advantage of planning nondeterministic chapters is the possibility of creating dramatic tension by means of planning to achieve weak goals, i.e. goals specifying that there is only a chance of achieving a certain situation.

4 Model for Generating Nondeterministic Plots

In order to increase the diversity of stories and the opportunities for interaction in Logtell, plan generation should fulfil the following requirements:

- **Support for nondeterministic events.** We need a structure to provide alternatives to the plot, so that the effects actually selected during dramatisation are taken into account and lead to the goals of the chapter.
- **Control on the level of nondeterminism.** The generated plans can, if many alternatives are chained, cause a combinatorial explosion on the number of possible continuations of a chapter and each continuation will have to be planned while the chapter is presented. In this way, it is necessary to be able to set a maximum value for the number of branches that can occur in the plan generated for a chapter.
- **Incorporation of the concept of "weak goals".** To create dramatic tension corresponding to attempts to establish a situation that can either succeed or fail, it is necessary to allow *weak goals*, which are considered fulfilled if there is a chance that the execution of the plan leads to a certain condition. It is also desirable to be able to establish how close of reaching that condition the plot should be. By incorporating weak goals, it is easier to specify that it is acceptable to finish a chapter with completely different alternatives.

- **Speed of plan generation.** The inclusion of events with nondeterministic effects results in a higher level of complexity during the planning stage. In addition, it is necessary to plan all possible continuations of a chapter. In this way, alternatives to speed up plan generation are crucial. It is however desirable that these alternatives do not limit the number of alternatives for the chapters.

In order to fulfil the requirements, the model considers that the planning job should be divided in two stages. In the first phase, there is a partial-order planning execution, in which, after incorporating user interventions or inferring new goals, a sketch of the solution is created. After that, an HTN-planning phase details the plan, creating a decision tree that takes into account the nondeterminism. The mix of partial-order planning and HTN planning accelerates the planning process, but at the same time does not restrict the solutions to previously specified HTN methods.

The sketch generated in the first phase is a partial-order plan containing both basic events and complex events (that correspond to complex HTN tasks). In this phase, the number of events should be limited to a small number in order to reduce the search space. Nondeterminism in this phase is taken into account to insert events that achieve weak goals. Each event has a list of deterministic effects and a list of alternatives of nondeterministic effects. A nondeterministic effect should be considered to establish a weak goal but not a strong goal. On the other hand, it should be taken into account that nondeterministic effects can affect the establishment of a strong goal. The fact that a complex event is an attempt to establish a certain situation can also be part of the effects of complex events, making easier the creation of the sketches.

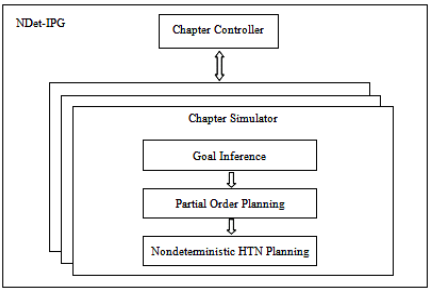


Fig. 1. Functional Schema of the NDet-IPG Planner

The second phase receives the sketch as an initial HTN to be detailed. It performs a forward-search execution, creating branches whenever a nondeterministic event is incorporated to the plan. Due to the treatment of weak goals by the planning phase, some events are marked as attempts. In the branches where their preconditions are not valid, they are discarded. The decision tree specifies the branch that should be chosen after each event. No matter the path of the generated decision tree, the initial HTN has to be accomplished.

The integration of the two planners constitutes a new module, called *Chapter Simulator*. To coordinate the activities between the *Chapter Simulator* and other modules of Logtell, a control module was created, the *Chapter Controller*, which is now responsible for generating the plot, chapter by chapter, including the treatment of

nondeterminism and the parallel processing of multiple valid alternatives at any given moment. Together, the Chapter Controller and the Chapter Simulator compose the *NDet-IPG* module (Figure 1), which now plays the role previously performed by IPG in Logtell`s architecture, being responsible for the generation of the plan to be used as input to the dramatisation process.

5 Implementation

Based on the current version of Logtell, we developed a prototype in Prolog capable of generating plots with nondeterministic events. This prototype is also able to, in accordance with the ideas outlined in the presentation of the model, deal with multiple planning jobs starting from different final states reached by a nondeterministic plan.

One of the modifications from the deterministic version of Logtell was in the literary genre definitions – more specifically, regarding the operators that specify the possible events. Now, each operator can have, in addition to a set of deterministic effects, alternative sets of nondeterministic effects.

The multiple planning jobs used to create the alternatives for the continuation of a chapter start from different states. To avoid the need for completely describing the current state with all its valid propositions for every action being incorporated, the adopted strategy was to keep only the changes made to the initial state. The effects established by the operators (and, consequently, by the actions that instantiate them) can provide positive facts (which must hold in the new produced state) and negative facts (which must not hold in the new state). Changes to the initial state are kept in two lists: one for the facts added to (*Facts+*) and one for the facts removed from (*Facts-*) the initial state. Every new action included in the plan has its effects (already properly instantiated) compared to these lists and to the initial state.

Most of the effort to implement the new plot generation mechanism was on the implementation of our HTN nondeterministic planner that performs the second phase of the planning process. The concepts of HTN subtasks and methods were implemented using a structure of inheritance and composition of operators. Apart from the primitive domain operators (which generate the effects in the current state, and are directly instantiated into actions), an operator can be either an abstraction of one or more specific operators and/or a composition of partially ordered sub-operators. In both cases, we call them *complex operators*. Abstraction relates a parent operator to one or more child operators. If the parent operator is also a composite operator, its children will inherit its composition. Children operators can also add new operators to the composition, and even include new operators amongst the inherited sub-operators.

During the second phase of the planning process, complex operators are replaced by their corresponding child and sub-operators, which can be not only domain operators, but also new abstractions or compositions - thus providing cascading inheritance and composition. Complex operators that are abstractions or compositions must be recursively instantiated or decomposed until domain operators are reached and incorporated to the plan. When a parent operator has more than one child operator, their preconditions will be evaluated by the planner in order to choose one of them, in the same way as an HTN method is chosen to accomplish a task.

As the HTN process performs a forward search procedure, whenever a primitive operator is selected, the nondeterministic planner updates the current state with its deterministic effects and, with this new current state, makes a new branch for each of its nondeterministic effects. These branches have their own current states updated with their respective nondeterministic effects, producing new current states which will be used to plan forward using the remaining task list. Thus, every time a nondeterministic operator is added to the plan, the planner will have new branches in its plan with their own sequences of operators and states. The greater the number of nondeterministic actions, greater will be the number of final states. In the decision tree, the conditions for the selection of each branch correspond exactly to the nondeterministic effects that generated it.

The adaptation of the partial-order planning of IPG was necessary to limit the number of operators the sketch to be generated can have and to incorporate the notion of weak goals. In order to deal with weak goals, the partial-order planner considers weak preconditions of the type *tried(LITERAL, N)*, where *LITERAL* is a fact or the negation of a fact and *N* specifies the tolerated number of events between a failed attempt and the moment when the weak precondition should be considered fulfilled. Whenever *tried(LITERAL, N)* is established by an operator, *tried(LITERAL, N_1)* is assumed to be established for any $N_1>N$. For the establishment of preconditions (or goals) of this type, it takes into account the operators that establish by means of their deterministic effects either explicitly *tried(LITERAL, N)* or just *LITERAL*. Moreover, if *LITERAL* is in one of the lists of nondeterministic effects of some event that surely occurs, then *tried*(*LITERAL*, 1) is considered to be established by this operator. The establishment of conditions of the type *tried(LITERAL, N)* with *N* greater than 1 can be done as if every precondition $PRECOND_i$ of an event that establishes (deterministically or not) *LITERAL* were rewritten as *tried($PRECOND_i$, N-1)*. In this case, the event should be considered an attempt of the type *ATTEMPT(N-1)*, which will be incorporated into the final plan if and only if its preconditions are reached. *ATTEMPT(0)* is an event that has to happen, i.e. all its preconditions must be deterministically established. Operators inserted as *ATTEMPT(N)* establish only effects (deterministic or not) of the type *tried(LITERAL, N+1)*. If any of these effects was already *tried(LITERAL, M)*, it must be changed to *tried(LITERAL, M+N+1)*.

In the second planning phase, the HTN planner only considers operators of the type *ATTEMPT* in the branches where all the operator's preconditions are met. When this is not the case, the operator is discarded. Effects and preconditions of the type *tried(LITERAL, N)* are not considered in the HTN planning phase. Goals of the type *tried(LITERAL, N)* appear in the inference rules (and/or in situations explicitly provided by the user) to increase dramatic tension and the opportunities for user interaction.

In order to test the prototype, we utilised the same storytelling context adopted in [3] (a small sub-class of the popular Swords and Dragons genre), with some modifications to include abstractions, compositions and nondeterministic effects. In this context, the events that can occur correspond to *attacks to the opponent's home*, *fights between characters*, *kidnapping of a victim*, *liberation of a victim*, *charms*, *weddings*, etc. Goal-inference rules used in this context establish relationships between situations and goals, such as: *if the villain is strong, the hero wants to become even stronger*; *if a victim is kidnapped, a hero will want to rescue her*; etc.

Our tests used mostly two nondeterministic operators, both of them being nondeterministic versions of operators from the original context used in Logtell (where all the operators were deterministic). The deterministic operator *kidnap(VIL, VIC)* has as its main effect that the character *VIC* (the victim) is kidnapped by the character *VIL* (the villain). It inspired the nondeterministic operator *try_to_kidnap(VIL, VIC)*, which has two possible outcomes: either the kidnapping succeeds and the result is exactly the same as in a deterministic kidnapping, or the villain ends up killing his victim instead of actually kidnapping her. Another operator from the original context is *fight(CH1, CH2)*, intended to model the situation where two characters fight with each other, having as its main consequence a decrement on their strengths. Its nondeterministic counterpart, the operator *fight_to_the_death(CH1, CH2)*, has two possible effects, each one corresponding to the death of one of the characters.

The operators have been remodelled to contemplate a hierarchy where one of the abstract operators, *perform_heroic_act(HER)* (used to make the hero do something "heroic"), can be specialised into two alternative complex operators: *rescue(HERO,VIC)* and *avenge(HERO,VIC)*, where the character *HERO* (the hero), rescues or avenges the character *VIC* (the victim), depending on the fact that *VIC* is alive or not. Both operators have the possibility of being decomposed into an alternative containing a fight to the death between the hero and the villain. If after the partial-order planning phase the initial task network has the operator *try_to_kidnap* preceding *perform_heroic_act*, the HTN planner is able to create a plot with the two nondeterministic events in sequence, each one with two possible results. As expected, the generated decision tree resulted in four different leaves, depending on the victim being killed or not, and on who is killed in the fight between the hero and the villain. Moreover, the decision tree accordingly deals with each situation after the attempt to kidnap the victim (either avenging her if she was killed, or rescuing her, otherwise).

Other tests have been made starting more than one task network (each one with its own initial state) concurrently. Even with six plans being generated together, with some of them having nondeterministic actions, we were able to verify the effects of HTN planning in the speed of the planning process. Preliminary experimental comparisons showed that the nondeterministic HTN planning process in NDet-IPG is faster than MBP planning on a similar context. The time to generate multiple alternative chapters in parallel is comparable to the time IPG used to take to generate a chapter without using HTN.

6 Concluding Remarks

Adding nondeterminism to the plots generated by Logtell is a powerful means to provide a higher level of interactivity with the user and to enhance the dramatic tension and the diversity of stories. Even if the user does not interact, the outcomes of any nondeterministic event can be chosen by chance, incrementing the replay value.

The experiments made so far showed that the model can cope with nondeterministic events. The use of HTN planning resulted in short planning times, even with a reasonable level of branching. However, some work is yet to be done. The binding with the other modules of Logtell is under implementation and is important to fully validate the user experience. In particular, dramatisation is also being adapted to deal

with nondeterminism and to allow the users to interfere in the events being dramatised, as described in [21]. In addition, we intend to look into the incorporation of models for the emotion of characters and the emotion transmitted by each event, so that they can be considered in the plot creation process.

References

1. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practice, 1st edn. Morgan Kaufmann Publishers, Amsterdam (2004)
2. Camanho, M.M., Ciarlini, A.E.M., Furtado, A.L., Pozzer, C.T., Feijó, B.: Conciliating Coherence and High Responsiveness in Interactive Storytelling. In: Proceedings of the 3rd ACM International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA), Athens (2008)
3. Ciarlini, A.E.M., Pozzer, C.T., Furtado, A.L., Feijó, B.: A Logic-Based Tool for Interactive Generation and Dramatization of Stories. In: Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2005), Valencia, pp. 133–140 (June 2005)
4. Ciarlini, A.E.M., Camanho, M.M., Dória, T.R., Furtado, A.L., Pozzer, C.T., Feijó, B.: Planning and Interaction Levels for TV Storytelling. In: 1st Joint International Conference on Interactive Digital Storytelling, Erfurt, Germany, November 26-29 (2008)
5. Erol, K., Hendler, J., Nau, D.S.: UMCP: A sound and complete procedure for hierachical task-network planning. In: Proceedings of the International Conference on AI Planning Systems (AIPS), pp. 249–254 (1994)
6. Cassandra, A.R., Kaelbling, L.P., Littman, M.L.: Acting optimally in partially observable stochastic domains. In: Proceedings of the Twelfth National Conference on Artificial Intelligence, Seattle, WA (1994)
7. Knight, F.H.: Risk, Uncertainty and Profit, Hart, Schaffner, and Marx (1921)
8. Cimatti, A., Pistore, M., Roveri, M., Traverso, P.: Weak, strong, and strong cyclic planning via symbolic model checking. Artif. Intell. 147(1-2), 35–84 (2003), http://www.informatik.uni-trier.de/%7Eley/db/journals/ai/ai147.html
9. Nau, D., Muñoz-Avila, H., Cao, Y., Lotem, A., Mitchell, S.: Total-Order Planning with Partially Ordered Subtasks. In: IJCAI 2001, Seattle (August 2001)
10. Kuter, U., Nau, D.S.: Forward-chaining planning in nondeterministic domains. In: AAAI (2004)
11. Kuter, U., Nau, D.S., Pistore, M., Traverso, P.: Task decomposition on abstract states, for planning under nondeterminism. Artif. Intell. 173(5-6), 669–695 (2009)
12. Riedl, M., Young, R.M.: From Linear Story Generation to Branching Story Graphs. IEEE Computer Graphics and Applications 26(3), 23–31 (2006), doi:10.1109/MCG.2006.56
13. Cavazza, M., Charles, F., Mead, S.: Character-based interactive storytelling. IEEE Intelligent Systems, Special Issue on AI in Interactive Entertainment 17(4), 17–24 (2002)
14. Charles, F., Cavazza, M., Mead, S.: Character-driven story generation in interactive storytelling. Technical report, VSMM, Berkeley (2001)
15. Young, R.: An overview of the mimesis architecture: Integrating narrative control into a gaming environment. In: Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, pp. 78–81. AAAI Press, Stanford (2001)
16. Propp, V.: Morphology of the Folktale, 2nd edn. University of Texas Press, Austin (2003)

17. Grasbon, D., Braun, N.: A morphological approach to interactive storytelling. In: Proc. CAST 2001, Living in Mixed Realities. Special Issue of Netzspannung.org/journal, the Magazine for Media Production and Inter-media Research, Sankt Augustin, Germany, pp. 337–340 (2001)
18. Spierling, U., Braun, N., Iurgel, I., Grasbon, D.: Setting the scene:playing digital director in interactive storytelling and creation. Computers & Graphics 26(1), 31–44 (2002)
19. Mateas, M., Stern, A.: Structuring content in the Facade interactive drama architecture. In: Proc. Artificial Intelligence and Interactive Digital Entertainment Conference, AIIDE (2005)
20. Crawford, C.: Assumptions underlying the Erasmatron storytelling system. In: Working Notes of the 1999 AAAI Spring Symposium on Narrative Intelligence. AAAI Press, Menlo Park (1999)
21. Dória, T.R., Ciarlini, A.E.M., Andreatta, A.: A Nondeterministic Model for Controlling the Dramatization of Interactive Stories. In: Proceedings of the ACM MM 2008 - 2nd ACM Workshop on Story Representation, Mechanism and Context - SRMC 2008, Vancouver, Canada (2008)

Diagnosis of Medical Images Using an Expert System

Itzel Abundez Barrera, Eréndira Rendón Lara,
Citlalih Gutiérrez Estrada, and Sergio Díaz Zagal

Lab. Reconocimiento de Patrones, Instituto Tecnológico de Toluca
Av. Tecnológico s/n, 52140 Metepec, Mexico
{iabundez,erendon,citlalihg,sdiaz}@ittoluca.edu.mx

Abstract. The possibility of developing systems to support medical diagnosis through Artificial Intelligence *(AI)* allows conceiving Expert Systems *(ES)*, which constitute successful methods to solve AI problems to program intelligent systems. This work deals with creating an *ES* to support the diagnosis of cervical lesions by identifying them in colposcopic images; for this purpose, 140 images were analyzed, with the most interesting and relevant result from this action being the definition of discriminating features: surface, color, texture and edges. These features will be used to evaluate an image and offer diagnosis as established by the expert physician, like: normal, inflammation process, immature metaplasia, gland eversion and low-grade lesion. To evaluate the system's performance, we obtained support from an expert colposcopy physician, who evaluated all 140 images. The results indicated that the *ES* obtained an efficiency of 75.75 % and an error percentage of 20.405%, including 4.04% that was not evaluated by the expert, who declared that the region or lesion was impossible to identify because the image was not clear.

1 Introduction

Artificial Intelligence is a very popular branch of Computer Sciences with a focus on enhancing the quality of certain aspects of our life by delegating activities to several types of machines, from car-assembling robots to disease-dignosing computer systems. This paper describes a research work diagnosing ROIs (Regions of Interest) through an Expert System *(ES)*, where a expert colposcopy physician analyzes a colposcopic image and answers the questions posed by the *ES*. By using these answers, the *ES* offers a medical diagnosis that is used by the specialist to diagnose cervical lesions in colposcopic images; the purpose of this process is to speed up medical diagnosis and increase timely detection of the aforementioned lesions. It may also be used as a cognitive teaching tool in colposcopy. Colposcopic images from the cervix and uterus areas are obtained with a special camera referred to as colposcope; gynecologists use these images to detect lesions in the area [1]. Lesions are considered signals used to diagnose several diseases, from the Human Papillomavirus *(HPV)* to cervical cancer [2]. Use of these images is very popular to treat patients; however, knowing and properly classifying them is an expensive process of complex learning for those initiating their work in this branch of medicine. As colposcopic examinations are extremely relevant, it is necessary to implement a supporting tool that provides reliable and useful analysis of colposcopic studies.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 144–152, 2010.

ES are knowledge-based systems, their potential based precisely on the information obtained from an expert in a specific area. Such knowledge is modeled after cause-effect rules obtained from the expert's experience and the public knowledge of the context. These are referred to as heuristic rules [3]. The expert's knowledge is obtained through the analysis of similar problems of successful outcomes and conclusions derived from failed attempts to solve the problem. Therefore, this work uses the expert's knowledge to develop a medical diagnosis system. This paper is organized as follows: Section 2 summarizes the art status. Section 3 describes the methodology used in the research work. Finally, section 4 reports the obtained results and experiences

2 Previous Works

An *ES* is a program or software endowed with certain intelligence that allows solving specific types of problems by simulating the human expert functions [4].

ES applications are as diverse as knowledge specialization areas, but may be classified depending on the specific function they carry out; in most cases, they solve problems, but the difference lies in the way solutions are approached. *ES* may be dedicated to interpret information or to identify models within a set of data. *ES* have found in medicine the best way to prove their potential by carrying out diagnosis comparable to those offered by novice physicians; some examples are the first diagnosis ES developed in the seventies [5] like: The medical diagnosis *ES* referred to as MYCIN, which allowed to diagnose and cure bacterial diseases [5], the system to diagnose pulmonary diseases called PUFF [5], another system developed to supervise intensive care patients called VM, etc. [3].Recent works as the one offered D. L. Hudson [6] "Use of neural network techniques in a medical expert system" Expert systems in medicine have relied heavily upon knowledge-based techniques, in which decision making rules or strategies are derived through consultation with experts. These techniques, coupled with methods of approximate reasoning, have produced systems which model the human decision making process. This approach has the disadvantage of requiring extensive interviewing of experts for each new application. It is desirable to be able to supplement this information by extracting information directly from data bases, without expert intervention. In this article, a neural network model is used to extract this information, and then use it in conjunction with rule-based knowledge, incorporating techniques of approximate reasoning.

Another related work is that of Rashid Jalal [7], who carried out a study to develop an automatic diagnosis system (or *ES*) to automate the interpretation of myocardial perfusion images using knowledge and heuristic approaches. The system correctly diagnoses obstructions at the three main arteries in patients with this condition, and the lack of obstructions in normal patients (without obstructed arteries). A third work proposed by R. Bellazzi [8] uses the Promedas (PRObabilistic MEdical Diagnostic Advisory System) model for internal medicine. This medical knowledge-based model consists of approximately 2000 diagnosis, 1000 results and 8600 connections between diagnosis and results, covering a large area of internal medicine. It demonstrated that Belief Propagation (BP) can be successfully applied as approximate algorithm of the Promedas network inference. It was therefore proved that BP constitutes an attractive alternative to accurate inference in the complex inference tasks of medical diagnosis.

The latter evidences that the medical field uses *ES* capabilities, and that they may be widely applied as long as there are problems whose solutions depend on data analysis and an expert that can support the *ES* creation process.

3 Methodology Description

An *ES* is basically constituted by three modules: inference machine, work memory and communication module; as the work memory includes the knowledge base and fact base, the author [4] consider that each base corresponds to a different module. To carry out the knowledge analysis and modeling, as well as implementation and tests, the methodology of Lezcano [9] was used; this methodology focuses on developing the *ES* with inference machine being the Prolog language. It has three stages:

- Identification: This is the stage to define the problem and its features, as well as the people that will intervene to make the knowledge base.
- Definition of Concepts: At this stage, the Knowledge Engineer and the expert set key aspects and the relations between them, as well as input and output data; these elements are grouped into classes of objects based on three basic principles: familiarity, adherence and scope.
- Formalization: Defined concepts are formalized to develop structures and ways to represent knowledge.

After formalization, the system's implementation is developed considering a grow- ing prototype approach. Prototypes are created, corrected and modified according to the expert criteria. There is a trial and tuning up period, in which the application's performance is evaluated.

The previous concept was the base to develop this research's *ES*. Key aspects of medical diagnosis were defined based on the process followed by the expert in colposcopy. Table 1 displays the diagnosis conditions; each column corresponds to a variable (characteristic) and rows represent the conditions formed by a group of columns, with which a specific result is obtained (diagnosis). Value 'X' is assigned to the characteristic that is irrelevant to provide a diagnosis.

Following Lezcano's methodology [9], during the first two stages (Identification and Definition of Concepts), the expert's normal work was observed, analyzing several cases in samples and identifying problem characteristics. With an approximation of discriminating characteristics, the result was verified with the expert and the analysis formalization was carried out later after the expert's approval.

According to the expert's diagnosis conditions, formalization was carried out by generating a decision tree.

As Prolog uses backwards chaining to find a solution, a useful tree or grid for problem representation is the Y-O tree. Figure 1 shows the tree used in this research.

The tree in Figure 1 depicts the Diagnosis to be obtained by the *ES*, which are: Normal, Inflammation Process, Immature Metaplasia, Mature Metaplasia, Gland Eversion and Low-Degree Lesion. It is therefore necessary to comply with the four sub-objectives to Analyze Images: surface, edges, color and texture. Note the arc below the Analyze Image circle joining the nodes edges with sub-objectives. The arc indicates that Analyze Image is a node called "Y" that may only be complied with if the four sub-objectives

Table 1. Conditions for Diagnosis

Diagnosis	Surface	Edge	Color	Texture
Normal	Smooth	No edges	Light pink	Even
	Smooth	No edges	Pink.	Even
	X	X	Pink	With cavities
Mature metaplasia	X	X	Dark pink	With cavaties
	X	X	Red	With cavaties
	X	X	Light pink	With cavaties
Immature metaplasia	X	X	Clear white	With cavaties
	X	X	white	With cavaties
Presents sections with edges				
Gland eversion		X	X	With glands
Low-degree lesion	X	X	White	Even

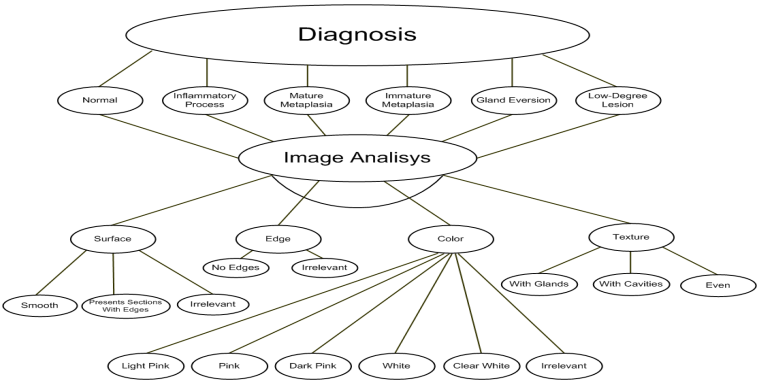


Fig. 1. Y-O tree used

joined with the arc are satisfied. Sub-objectives without arc are nodes called ”O”, in which the satisfaction of any of these sub-objectives complies with the parent objective. The operation of the *ES* developed has a procedure based on the questions made to the user, through which a result is obtained. These questions are associated to the appearance of the image observed. For the *ES* to offer a result, the following process must be carried out:

- Video acquisition: This first stage requires the use of a colposcope that sends a video signal, which is then captured by the computer and stored in a file for later analysis. Once the video is stored, a scene from such file is selected. The scene selection process is based on the region of interest, considering the criteria of the most accurate image.
- Image Analysis: The user-selected image is used to pinpoint one or several regions of interest. Then, four questions about the appearance of the region are made; these questions are as follows:

1. Image surface: The type of surface observed in the region of interest is selected; surfaces may be: Smooth or Sections with edges.
2. Type of edges: If the region of interest has edges, the user will specify their type, for example: Irregular or Regular.
3. Color of Region of Interest: The color of the region of interest is chosen; op tions may be: Clear White, White, Light Pink, Pink, Dark Pink, Red.
4. Texture of Region of Interest: The region texture of interest is chosen; options may be: with cavities, with glands, even. Figure 2 depicts the system when the expert is selecting ROIs.

– Issuance of Results - The user requests the analysis of the region of interest and the system uses an inference process (inference is the reasoning process carried out by the *ES*, similar to the process carried out by an expert to deduce the solution of a problem) to get the Diagnosis.

Fig. 2. Image Analysis

Figure 3 displays the system when issuing a Diagnosis. The *ES* is a Hybrid System implemented with Java and Amzi Prolog that uses Prolog as the *ES* inference machine and Java to develop the graphic interface. Figure 4 depicts the use case diagram to develop the *ES*. With the *ES*, it is possible to support the colposcopy specialist in issuing a medical diagnosis. This *ES* is the base to extract knowledge from the human expert and continue with a larger project, which is the automatic identification of cervical-uterine lesions.

4 Results

The sample used for our experiments was taken from the database of the Mother and Child Hospital of the Mexican State, constituted by 140 images, which were previously identified and analyzed by the colposcopy expert. Each image corresponds to different patients and each image may offer one or more regions of interest. Considering the latter, a total sample of 198 regions of interest was obtained and such regions have a diagnosis previously defined by the expert specialist. Figure 5 presents the diversity of regions of interest in the sample.

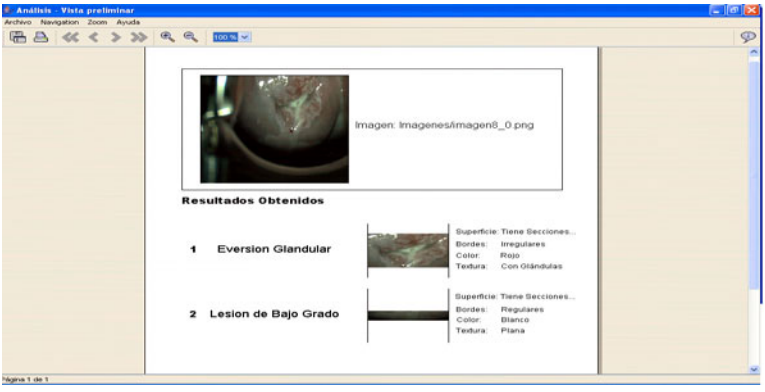


Fig. 3. Diagnossis

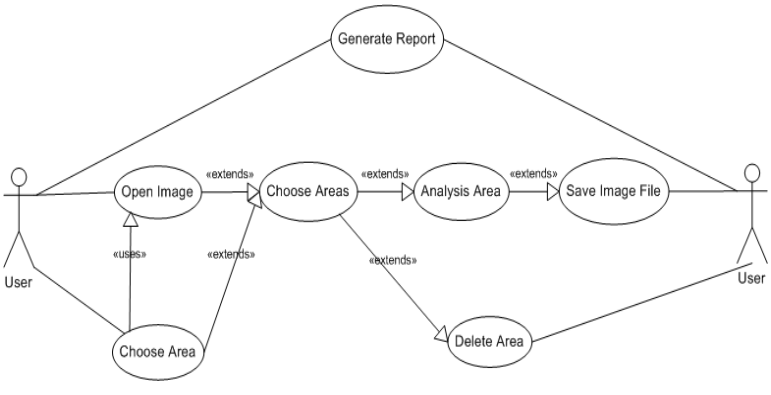


Fig. 4. ES Used Case diagram

The tool used to present and analyze the diagnosis results is the matrix of confusion, also known as contingency matrix; it presents the correct percentage of diagnosis and the reliability of each class and, most importantly, confusions between classes.

Table 2 offers the data that the Immature Metaplasia Diagnosis has 39 regions of interest, which is the diagnosis that generated the highest number of confusions, as follows: one confusion was diagnosed as Normal, two regions were diagnosed as Inflammatory Process, three as Mature Metaplasia and two as Gland Eversion, resulting in a correct diagnosis of 72.972 % for this class. The Normal Class obtained the highest percentage of correct diagnosis, with 92.105 %, and there were three confusions, two of them with Inflammatory Process, and one with Immature Metaplasia, and so on for each diagnosis.We used next name columns: A = Normal;B =Inflammatory Process;C = Mature Metaplasia;D = Immature Metaplasia; E =Gland eversion; F =Low-degree lesion for Table 2. Figure 6 depicts incidences of the different confusions that occurred. Eight of the regions of interest equivalent to 4 % were labeled as ”impossible to diagnose” because in images, the region of interest was not clear enough, and the perception

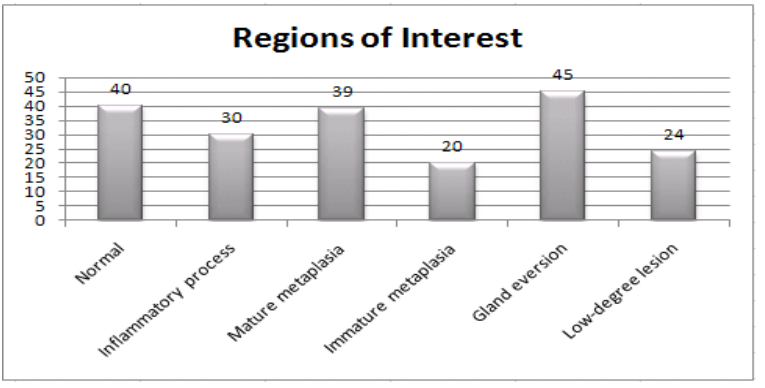


Fig. 5. Diversity of the sample

Table 2. Confusion Matrix

Diagnosis	A	B	C	D	E	F	Misclassified	Allocation class	Efficiency
A	35	0	0	0	1	0	4	40	87.7
B	2	22	3	1	1	1	0	30	73.33
C	1	2	27	3	2	1	3	39	69.23
D	0	3	0	14	2	1	0	20	70
E	0	3	5	3	34	0	0	45	75.55
F	1	2	0	2	18	1	24	75	
Overall	38	31	37	21	42	21	8	198	
%Efficiency	92.10	70.96	72.97	66.66	80.95	85.71			

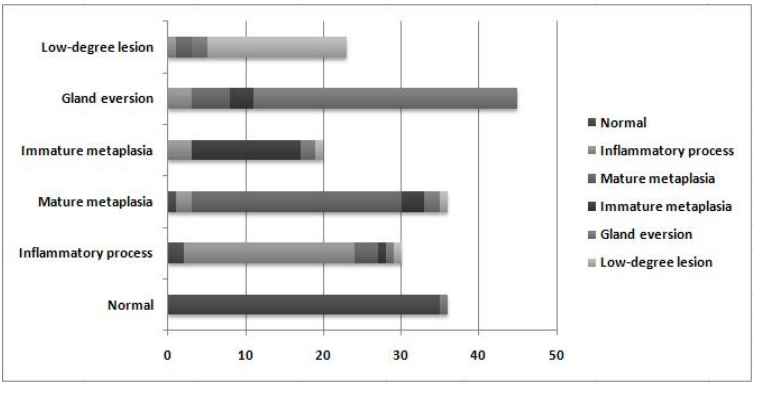


Fig. 6. Incidences

of the expert when analyzing the region of interest, through the use of the colposcope, had better definition than that obtained when observing the image in the system. From a total of 198 regions of interest, 150 were correctly diagnosed, corresponding to 75.75%

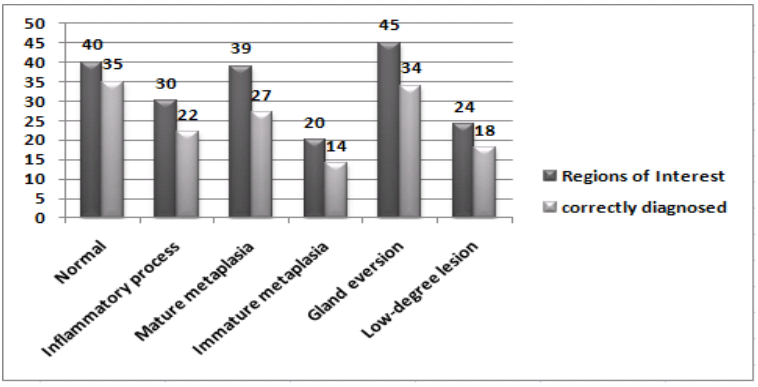


Fig. 7. Performance per areas

of positive markings, 20.25% of error and 4 % declared as ”impossible to classify”. Figure 7 offers the performance obtained at each of the sample’s diagnosis.

5 Concluding Remarks

The *ES* performance in the tests was 75.75 % efficient in the Diagnosis of the sample’s different regions of interest; one of the reasons is that the expert’s perception when analyzing the region of interest through the colposcope is different to that obtained when observing the image in the system. That is, regions have better definition when seen from a colposcope than when seen from a digital image.

ES are a powerful tool in this work to support the diagnosis and represent knowledge in the form of rules, easing the development of the application and the use of an inference machine.

The contribution of this research is to extract knowledge from the expert and under- stand the way in which he/she makes diagnosis to reflect knowledge from the beginning in an *ES*, so to allow the development of an automatic diagnosis system in the future.

Considering the percentage of positive marks obtained with the system, it is still necessary to run several tests to decrease the error percentage because, with these val- ues, the system cannot be implemented. However, if we consider some of the results from other systems, the ES is within acceptable values, even with an error of 20.25 %. This is a motivation to keep working and enhance the system’s efficiency.

This work is supported by the information offered by the evaluation of colposcopy experts from the Mother Child Hospital of the Institute of Social Security of the State of Mexico (ISSEMyM) , within the research project ”Diagnosis of Cervical-Uterine Lesions Through Pattern-Recognition Techniques”; therefore, the information is true and reliable.

Acknowledgments

We would like to thank especialist. Irma Ma. de los Ángeles Alonso Reyna, Head of the Clinic of Dysplasia of the Mother Child Hospital of the ISSEMYM, who collaborated in the work of analyzing and identifying videos and colposcopic images.

We would also like to express our gratitude to DGEST for the support granted to the project "Diagnosis of Cervical-Uterine Lesions Through Pattern-Recognition Techniques", registered as 555.07-P.

References

1. Apgar, B.S.: Colposcopic Principles and Practice. McGrawHill, México (2003) (in Spanish)
2. Coupez, F.: Initiation à la Colposcopie. MASSON, Paris (1990)
3. Rusell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Pearson-Prentice Hall, London (2002)
4. Peter, J.: Introduction to Expert Systems, 3rd edn. Addison-Wesley, England (1999)
5. Joseph, G., Gary, R.: Expert Systems:Principles and Programming, 3rd edn. Thomson (1991)
6. Hudson, D.L., Cohen, M.E., Anderson, M.F.: Use of neural network techniques in a medical expert system. International Journal of Intelligent Systems 6(2), 213–223 (2007)
7. Qureshi1, R.J., Husain, S.A.: Design of an Expert System for Diagnosis of Coronary Artery Disease Using Myocardial Perfusion Imaging. In: National Conference on Emerging Technologies (2004)
8. Bellazzi, R., Abu-Hanna, A., Hunter, J.: Inference in the Promedas Medical Expert System. In: Bellazzi, R., Abu-Hanna, A., Hunter, J. (eds.) AIME 2007. LNCS (LNAI), vol. 4594, pp. 456–460. Springer, Heidelberg (2007)
9. Lezcano Brito Mateo G.: Prolog and Expert Systems. Pandora, Mexico (2000) (in Spanish)

Revision of CTL Models

Paulo T. Guerra and Renata Wassermann

Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo, Brazil
{paulotgo,renata}@ime.usp.br

Abstract. Model checking is one of the most robust techniques in automated system verification. But, although this technique can handle complex verifications, model checking tools usually do not give any information on how to repair inconsistent system models. In this paper, we show that approaches developed for CTL model update cannot deal with all kinds of model changes. We introduce the concept of CTL model revision: an approach based on belief revision to handle system inconsistency in a static context. We relate our proposal to classical works in belief revision and give an algorithm sketch.

1 Introduction

Model checking is an efficient method for automated verification of systems specification. Proposed by Clark and Emerson [1], this technique allows a system designer to check whether a system model satisfies (or not) its formal specification, i.e., the set of desired properties that defines the system behavior. In the last decades, many studies as [2–4] have improved model checking, resulting in fast algorithms and, consequently, in the development of efficient verification tools.

Most of these tools just return a counterexample path when they find an inconsistency. This information is useful for system designers, because they can direct the correction to a specific point of the system. However, the more complex a system is, the harder it becomes for the designer to fix it. In general, model checking tools do not have any mechanisms to assist the user in the correction task.

To address this issue, it is quite natural to think of integrating model checking and some automated change technique. The work [5] was the first in this line. Buccafurri et al. integrate model checking and abductive theory revision to perform system repairs. Although this approach can be successfully applied to their purposes, the authors themselves state that it may not be general enough to deal with other system modifications.In [6], Zhang and Ding propose a formal framework for model update through the integration of model checking and belief update [7–9]. They specify a minimal change principle for model update, and then define the concept of admissible update. They also specify a model update operator and study its semantics and computational properties.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 153–162, 2010.

While belief update deals with changing information due to changes in the world, belief revision [10] deals with incoming information about a static world. Despite the similarity of the problems, the use of an incorrect type of change can lead to information loss.

In our work, we propose the use of belief revision for changing CTL specifications. Our approach addresses the repair of systems specifications in a static context. As Zhang and Ding do to update, we define a model revision operator and characterize its relationship with some classical work on belief revision.

2 CTL Model Checking

Formal verification methods check whether a system model, specified in a formal language, satisfies a set of desired properties. Model checking [11, 12] is probably the most widely used verification method. According to [11], model checking is defined as the process of computing an answer to the question whether $(M, s) \models \phi$ holds, where ϕ is a temporal formula, M is an appropriate model of the system under consideration, s is a state of that model, and $\models$ is the underlying satisfaction relation.

In this paper, we will address only a special type of model checking, the *CTL model checking*. We will assume that systems are described by Kripke models and that the desired properties are CTL formulas. We introduce both concepts below.

2.1 Computation Tree Logic

Temporal logic is a kind of modal logic where we can represent and reason about time sentences. *Computation Tree Logic* [11, 12], CTL for short, models the future as a tree-like structure. Due to its branching characteristic, CTL can quantify over several execution paths that a system can follow. In Definitions 1 and 3, we briefly show the CTL syntax and semantics, respectively (see [12] for details).

Definition 1 (Syntax). *A CTL formula has the following syntax:*

$$\varphi ::= \top \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid EX\varphi \mid EG\varphi \mid E[\varphi U \varphi]$$

where p ranges over a set of atomic formulas, $\neg$, $\wedge$ are the classical logic connectives, and the remaining elements are temporal operators.

We use $\bot$ to denote $\neg\top$, and other temporal operators can be derived from EX, EG and EU: AX $\phi = \neg$ EX $\neg\phi$; AG $\phi = \neg$ EF $\neg\phi$; AF $\phi = \neg$ EG $\neg\phi$; EF $\phi =$ E [true U ϕ]; A$[\phi$ U $\beta] = \neg$ E$[\neg\beta$ U $\neg\phi \wedge \neg\beta] \wedge \neg$ EG $\neg\beta$.

Each temporal operator consists of a path quantifier (E, "there is a path", or A, "for all paths") followed by a state operator (X, "next state", U, "until", G, "globally in states" or F, "some future state").

Definition 2. *Let P be a finite set of atomic propositions, a* Kripke model *is a triple $M = (S, R, L)$, where*

1. *S is a set of states*
2. *$R \in S \times S$ is a serial relation between states, that is, $\forall s \in S$, $\exists s' \in S$ such that $(s, s') \in R$.*
3. *$L : S \to 2^P$ is a labeling function such that $\forall s \in S$, $L(s)$ determines all atomic propositions true in s.*

Definition 3 (Semantic). *Let $M = (S, R, L)$ be a Kripke model, $s \in S$ a state of M and ϕ a CTL formula. We define $(M, s) \models \phi$ inductively as follows:*

1. $(M, s) \models \top$.
2. $(M, s) \models p$ *iff* $p \in L(s)$.
3. $(M, s) \models \neg\phi$ *iff* $(M, s) \not\models \phi$.
4. $(M, s) \models \phi_1 \wedge \phi_2$ *iff* $(M, s) \models \phi_1$ *and* $(M, s) \models \phi_2$.
5. $(M, s) \models EX\phi$ *iff* $\exists s' \in S$ *such that* $(s, s') \in R$ *and* $(M, s') \models \phi$.
6. $(M, s) \models EG\phi$ *iff there is a path* $\pi = [s_0, s_1, \ldots]$ *in* M *such that* $s_0 = s$ *and* $(M, s_i) \models \phi$ *for all* $i \geq 0$.
7. $(M, s) \models E[\phi_1 U \phi_2]$ *iff there is a path* $\pi = [s_0, s_1, \ldots]$ *in* M *such that* $s_0 = s$, $\exists i \geq 0$, $(M, s_i) \models \phi_2$ *and* $\forall j < i$, $(M, s_j) \models \phi_1$.

Given a CTL Kripke model $M = (S, R, L)$ and its set of initial states $Init(S) \subseteq S$, we say that $M \models \phi$, if and only if $(M, s) \models \phi$ for some $s \in Init(S)$.

3 Belief Revision

Belief revision deals with the problem of adapting an agent's set of beliefs in order to incorporate new information, possibly inconsistent with what the agent used to believe. In [10], Alchourrón, Gärdenfors and Makinson proposed a set of postulates that belief change should obey, as well as some constructions for revision that follow these postulates. This became known as the AGM paradigm.

In the AGM Paradigm, beliefs are represented as *belief sets*, that is, a set of formulas K such that $Cn(K) = K$, where Cn is a supraclassical consequence operator. Two other belief representation are important to our work: *belief bases* and *possible worlds*. When we use *belief bases* we assume that belief sets can be inferred from a small set of beliefs. A belief base B_K represents the belief set K such that $K = Cn(B_K)$. In a *possible worlds* representation, we do not represent the beliefs of the agent by sets formulas, but by all the models (worlds) that satisfy the beliefs. We denote by $[K]$ (some times $Mod(K)$) the set of models where every formula of a belief set K holds.

The AGM postulates guide revision operations through a minimal change principle, that is, nothing should be added or removed from a belief set if it is not strictly necessary to the success of revision. Given a belief set K, a belief α, $K+\alpha$ is the result of expanding K with α, usually defined as $K+\alpha = Cn(K \cup \{\alpha\})$. The operation of revision, denoted by $K * \alpha$ is not uniquely defined, but constrained by rationality postulates. Here, we will follow the notation adopted in [13], where the authors suggested a reformulation of AGM postulates in which a belief set is represented by a single sentence ψ and the result of revising ψ by ϕ is denoted by $\psi \circ \phi$.

(R1) $\psi \circ \phi \vDash \phi$.
(R2) If $\psi \wedge \phi$ is satisfiable, then $\psi \circ \phi \equiv \psi \wedge \phi$
(R3) If ϕ is satisfiable, then $\psi \circ \phi$ is satisfiable.
(R4) If $\psi_1 \equiv \psi_2$ and $\phi_1 \equiv \phi_2$, then $\psi_1 \circ \phi_1 \equiv \psi_2 \circ \phi_2$.
(R5) $(\psi \circ \phi) \wedge \mu \vDash \psi \circ (\phi \wedge \mu)$
(R6) If $(\psi \circ \phi) \wedge \mu$ is satisfiable, then $\psi \circ (\phi \wedge \mu) \vDash (\psi \circ \phi) \wedge \mu$

4 Belief Update

The purpose of *belief update*, as *belief revision*, is to preserve the consistency of belief sets when new information is added. While *belief revision* deals with static worlds, belief update acts on dynamic worlds, where the new information describes changes that occur in that world. This is the fundamental distinction between them: the nature of the change.

In belief revision, the new information means a refinement of beliefs, or even that the current belief set can not describe the world correctly. But in belief update, the new belief shows a change, it does not say anything about what the world was, just about what it looks like after the change.

Suppose we want to change the belief base ψ by a formula ϕ. Revision operators will select from the models of ϕ those that are in some sense closest to the models of ψ. On the other hand, update operators select, for each model M of ψ, the models of ϕ that are closest to M. Example 1 illustrates this distinction between revision and update.

Example 1. Suppose there is a room containing three objects: a book, a magazine and a table. The proposition b means "the book is on the table" and the proposition m means "the magazine is on the table". We believe that $\psi \leftrightarrow (b \wedge \neg m) \vee (\neg b \wedge m)$, that is, either the book or the magazine are on the table, but not both. There are only two models: I, b is true and m is not; or J, where m is true and b is not.

Suppose now that we send a robot to put the book on the table. After the success of the operation, we want to incorporate to our belief base the new belief $\phi \leftrightarrow b$. This belief also has two possible models: I and a new one K, where both, book and magazine, are on the table. We will select those models of ϕ which are closest to the models of ψ. We will assume as distance relation the number of proposition with different true value. The model I is at distance 0 of itself and at distance 2 of J (both propositions change). On the other hand, the model K is a distance 1 of both I and J (matching exactly one object location).

A revision operator will select just model I because it is the closest model to the original set. But intuitively this should not be the case. After the robot performs its task, all we know is that the book is on the table, but we know nothing about the magazine location.

Katsuno and Mendelzon [9] proposed a set of postulates to characterize rational update operations. Let ψ represent a finite belief base, ϕ the new belief and $\psi \diamond \phi$ the result of the update, the eight update postulates defined by Katsuno and Mendelzon are:

(U1) $\psi \diamond \phi \vDash \phi$.
(U2) If $\psi \vDash \phi$, then $\psi \diamond \phi \equiv \psi$.
(U3) If both ψ and ϕ are satisfiable, then $\psi \diamond \phi$ is also satisfiable.
(U4) If $\psi_1 \equiv \psi_2$ and $\phi_1 \equiv \phi_2$, then $\psi_1 \diamond \phi_1 \equiv \psi_2 \diamond \phi_2$.
(U5) $(\psi \diamond \phi) \wedge \mu \vDash \psi \diamond (\phi \wedge \mu)$
(U6) If $\psi \diamond \phi_1 \vDash \phi_2$ and $\psi \diamond \phi_2 \vDash \phi_1$, then $\psi \diamond \phi_1 \equiv \psi \diamond \phi_2$
(U7) If ψ is complete (i.e has a unique model) then $(\psi \diamond \phi_1) \wedge (\psi \diamond \phi_2) \vDash \psi \diamond (\phi_1 \vee \phi_2)$
(U8) If $(\psi_1 \vee \psi_2) \diamond \phi \equiv (\psi_1 \diamond \phi) \vee (\psi_2 \diamond \phi)$

Herzig [14] shows that, more than distinct, revision and update are mutually incompatible, since there is not an operator able to satisfies both revision and update postulates.

5 CTL Model Change

Let us now depart from propositional logic and see what happens when we want to change formal specifications, where models are Kripke models and the input is a CTL formula. First we will look at changing a single model $M = (S, R, L)$. Zhang and Ding [6] defined five basic operations where all possible changes on CTL models could be achieved.

PU1 : Adding one pair to the relation R
PU2 : Removing one pair from the relation R
PU3 : Changing the labeling function on one state
PU4 : Adding one state
PU5 : Removing one isolated state

Given two CTL Models $M = (S, R, L)$ and $M' = (S', R', L')$, for each PUi $(i = 1, \ldots, 5)$, we denote by $Diff_{PUi}(M, M')$ the difference between M and M', where M' is an updated model from M, where

1. $Diff_{PU1}(M, M') = R' - R$ (the set of pairs added to the relation).
2. $Diff_{PU2}(M, M') = R - R'$ (the set of pairs removed from the relation).
3. $Diff_{PU3}(M, M') = \{s \mid s \in S \cap S' \text{ and } L(s) \neq L'(s)\}$ (the set of states whose labeling function has changed).
4. $Diff_{PU4}(M, M') = S' - S$ (the set of added states).
5. $Diff_{PU5}(M, M') = S - S'$. (the set of removed states).

Based on these metrics, Zhang and Ding define the ordering $\leq_M$: a measure on the difference between two CTL models with respect to a given model.

Definition 4 (Closeness ordering). *Let $M = (S, R, L)$, $M_1 = (S_1, R_1, L_1)$ and $M_2 = (S_2, R_2, L_2)$ be three CTL Kripke models. We say that M_1 is at least as close to M as M_2, denoted as $M_1 \leq_M M_2$, if and only if for each set of PU1-PU5 operations that transform M in M_2, there exists a set of PU1-PU5 operations that transform M in M_1 such that the following conditions hold:*

1. *for each* i $(i = 1, \ldots, 5)$, $Diff_{PUi}(M, M_1) \subseteq Diff_{PUi}(M, M_2)$, *and*
2. *if* $Diff_{PU3}(M, M_1) = Diff_{PU3}(M, M_2)$, *then for all* $s \in Diff_{PU3}(M, M_1)$, $diff(L(s), L_1(s)) \subseteq diff(L(s), L_2(s))$, *where* $diff(A, B) = (A - B) \cup (B - A)$ *for any two sets* A *and* B.

We denote $M_1 <_M M_2$ *if* $M_1 \leq_M M_2$ *and* $M_2 \not\leq_M M_1$.

Having the ordering relation specified in Definition 4, the authors formally specify in Definition 5 what an admissible model update is:

Definition 5 (Admissible update). *Given a CTL model* $M = (S, R, L)$, $\mathcal{M} = (M, s_0)$ *where* $s_0 \in S$, *and a CTL formula* ϕ, *a CTL Kripke model* $Update(\mathcal{M}, \phi)$ *is called an admissible model (or admissible update model) if the following conditions hold:*

1. $Update(\mathcal{M}, \phi) = (M', s'_0)$, $(M', s'_0) \vDash \phi$, *where* $M' = (S', R', L')$ *and* $s'_0 \in S'$.
2. *there does not exist another updated model* $M'' = (S'', R'', L'')$ *and* $s''_0 \in S''$ *such that* $(M'', s''_0) \vDash \phi$ *and* $M'' <_M M'$.

$Poss(Update(\mathcal{M}, \phi))$ *denotes the set of all possible admissible models of updating* $\mathcal{M}$ *to satisfy* ϕ.

The main contribution of Zhang and Ding's paper is perhaps the semantical analysis of CTL model update. They analyze their approach with respect to classical belief update approaches, identifying common issues between them. Although they address essentially different problems, the notion of minimal change in CTL model update is closely related to what occurs in classical approaches.

To make possible a comparison between CTL model update and traditional update postulates, Zhang and Ding define an operator $\diamond_c$ based on Winslett's Possible Model Approach [15], but designed for CTL model update. The formal definition of $\diamond_c$ is given in Definition 6.

Definition 6. *Let* $M = (S, R, L)$ *be a CTL Kripke model,* $Init(S) \subseteq S$ *the set of initial states of* M *and* ϕ *a CTL formula.* M *is called a model of* ϕ *if and only if* $(M, s) \vDash \phi$ *for some* $s \in Init(S)$. *Denote by* $Mod(\phi)$ *the set of all models of* ϕ. *The model update operator* $\diamond_c$ *is defined as:*

$$Mod(\psi \diamond_c \phi) = \bigcup_{(M,s) \in Mod(\psi)} Poss(Update((M, s), \phi))$$

Zhang and Ding ([6], Theorem 1) proved that operator $\diamond_c$ satisfies all Katsuno and Mendelzon update postulates. They show that KM postulates can characterize a wider scope of update operations than just its traditional application on propositional logic. In this sense, Zhang and Ding show that KM postulates might be essential for any model based update approach.

6 CTL Model Revision

The need for CTL model revision is motivated through the fundamental difference between belief revision and belief update: the type of problems that each one is suitable for. As seen in Section 4, belief update is used to modify a belief set in order to accommodate a new information about a dynamic world, while belief revision is used when the new information is about a world that has not changed.

To illustrate a situation where belief revision is more adequate, let us go back to Example 1. In this example we believe that, in a given room, either a book is on a table (b) or a magazine is (m), but not both, that is, $\psi \leftrightarrow (b \wedge \neg m) \vee (\neg b \wedge m)$ where ψ represents our knowledge base. Then, we ask a robot to put the book on the table. In other words, we want to incorporate to our beliefs a new information, $\mu \leftrightarrow b$. An update operator generates a new knowledge base $\psi' \leftrightarrow b$, while a revision operator generates the knowledge base $\psi'' \leftrightarrow (b \wedge \neg m)$. Intuitively, the update result has a more rational behavior, because the whole situation does not give us any information about the magazine location.

But suppose that instead of asking the robot to put the book on table, we ask it to just enter in the room and tell us where the book is. Suppose now that when the robot comes back it informs us that the book is on the table. Note that the new belief remains the same, $\mu \leftrightarrow b$, and the results of update and revision operations will be identical to the previous case. But, in this new situation, the result of revision seems more appropriated.

According to the new information $\mu \leftrightarrow b$, two room configurations are possible: (1) the book is on the table and the magazine is on the floor ($b \wedge \neg m$) or (2) both, book and magazine, are on the table ($b \wedge m$). In the last case, we used to believe that book and magazine are not in the same location and we know that nothing in the room has changed, so there is no reason to us to have doubts about where the magazine is. We can see that revision produces a more informative result.

Note that, in both cases, the knowledge base ψ and the new information μ are identical, what changes is how a individual interacts with the world, and, therefore, the way that the agent should realize it. We believe that the same distinction problem between static and dynamic worlds should be applied to system specifications.

There are two main reasons to modify a system specification: (1) when it fails to satisfy a given requisite and (2) when it needs to be adapted to a new requisite. When we modify a specification motivated by (1), we believe that a repair approach based on belief revision could generate more informative results than that produced by the CTL Model Update approach, in a similar way to what occurs between belief revision and belief update.

An approach based on belief revision for handling inconsistencies was presented in [16, 17]. The authors enrich the NuSMV model checker [4] with principles of belief revision, which results in *BrNuSMV*[1], a tool capable of suggesting modifications to a given system specification such that some desired CTL property becomes true.

Perhaps because BrNuSMV is a initial step into the integration of model checking and belief revision principles, there are some gaps: it is not possible to deal with all CTL formulas; the revision process is done state by state, with restricted set of beliefs; there is no formal analysis of the belief revision process. In our work, we intend to fill some of these gaps, doing a formal analysis of a revision method applied to generic CTL models and studying its relation with classical belief revision approaches.

[1] Acronym for Belief Revision NuSMV.

We specify in Definition 7 our revision operator $\circ_c$ over CTL formula.

Definition 7. *Given two CTL formulas ψ and ϕ, we define that $\psi \circ_c \phi$ results in a CTL formula whose models are defined as*

$$Mod(\psi \circ_c \phi) = Min_{Mod(\psi)}(Mod(\phi))$$

where $Min_{\mathcal{B}}(\mathcal{A})$ denotes the set of all minimal models of $\mathcal{A}$ with respect to all orderings $\leq_M$ such that M is a model of $\mathcal{B}$.

Theorem 1. *Operator $\circ_c$ satisfies revision postulates (R1)-(R6).*[2]

In Algorithm 1, we present an algorithm sketch for CTL model revision. Given a belief base ψ and a new belief ϕ, both represented by CTL formulas, the revision algorithm computes a set $\mathcal{S}$ of CTL Kripke models such that for all $M \in \mathcal{S}, M \in Mod(\psi \circ_c \phi)$.

Algorithm 1. CTLModelRevision(ψ, ϕ)

Input: Two CTL formulas ψ and ϕ, representing the belief base and the new belief, respectively.
Output: A set of CTL Kripke models $\mathcal{S}$ such that $\mathcal{S} \subseteq Mod(\psi \circ_c \phi)$.

```
1: S ← ∅;
2: for all models M = (S, R, L) such that M ∈ Mod(ψ) do
3:    if M ⊨ φ holds then
4:       S ← S ∪ {M};
5:    else
6:       repeat
7:          S ← S ∪ CTLUpdate((M, s), φ), where s ∈ Init(S);
8:       until there is no change in S.
9: for all models M = (S, R, L) such that M ∈ Mod(ψ) do
10:   S ← S − {M'|{M', M''} ⊆ S and M'' <_M M'};
11: return S.
```

First, we initialize $\mathcal{S}$ as empty. In lines 2-8, we iterates over ψ's models to find models that satisfies ϕ. If a model M of ψ satisfies ϕ, then M is added to $\mathcal{S}$, otherwise, in lines 6-8, we look for a set of possible models that satisfy ϕ and are close to M. For this purpose, we make successive calls to $CTLUpdate((M, s), \phi)$, a function described in [6]. In lines 9-10, we remove from $\mathcal{S}$ all models that are not minimal in relation to ψ's models. After this step, all models in $\mathcal{S}$ satisfy ϕ and are minimal with respect to $\leq_M$. Finally, we return $\mathcal{S}$.

Zhang and Ding [6], Theorem 8, proved that $CTLUpdate((M, s), \phi)$ always generates a model M' that satisfies ϕ and is a closest model of M according to Definition 4. However, there are no guarantees that $CTLUpdate((M, s), \phi)$ generates all possible minimal models, so we can not ensure that our algorithm can obtain $Mod(\psi \circ_c \phi)$ as result.

Zhang and Ding [6] do not define a CTL model update algorithm. In order to compare the revision and update model approaches, we specify in Algorithm 2 a

[2] The proof is omitted due to space limitation.

model update method based in their $\diamond_c$ update operator. The difference between these two approaches is illustrated in Example 2.

Algorithm 2. CTLModelUpdate(ψ, ϕ)

Input: Two CTL formulas ψ and ϕ, representing the belief base and the new belief, respectively.
Output: A set of CTL Kripke models $\mathcal{S}$ such that $\mathcal{S} \subseteq Mod(\psi \diamond_c \phi)$.

```
S ← ∅;
for all models M = (S, R, L) such that M ∈ Mod(ψ) do
  if M ⊨ φ holds then
    S ← S ∪ {M};
  else
    repeat
      S ← S ∪ CTLUpdate((M, s), φ), where s ∈ Init(S);
    until there is no change in S.
return S.
```

Example 2. Let p and q be two propositional atoms and $\psi \leftrightarrow EXAGp$ our belief base. Suppose we need to modify ψ in order to accept $\phi \leftrightarrow E[pUq]$. Suppose also that ϕ does not describe a change in the world, just a new information about it. We know that $M_1 = (\{s_0, s_1\}, \{(s_0, s_1), (s_1, s_1)\}, \{L(s_0) = \{p\}, L(s_1) = \{p\}\}$ and $M_2 = (\{s_0, s_1\}, \{(s_0, s_1), (s_1, s_1)\}, \{L(s_0) = \{q\}, L(s_1) = \{p\}\}$ belong to $Mod(\psi)$. When we submit ψ and ϕ to CTL model update, we obtain a model $M_1' = (\{s_0, s_1, s_2\}, \{(s_0, s_1), (s_1, s_1), (s_1, s_2)\}, \{L(s_0) = \{p\}, L(s_1) = \{p\}, L(s_2) = \{q\}\}$, which does not belong to the output of the CTL model revision algorithm. Although M_1' is minimal in relation to M_1, M_1' it is not minimal in relation to all ψ's models (M_2 itself satisfies ϕ and it is $M_2 <_{M_2} M_1'$), so M_1' is not in $Mod(\psi \circ_c \phi)$, and it is removed in last loop of the revision algorithm.

7 Conclusion

We have shown that belief revision can be used to handle inconsistencies in formal system specifications when they are specified as CTL Kripke models. Based on the well-known difference between belief revision and belief update, we argue that our approach can be more adequate than Zhang and Ding's CTL model update when applied to system modifications in a static context. We have defined a revision operator for CTL and shown that our operator obeys the (R1)-(R6) revision postulates which shows that our operator performs revision in a rational way. Based on this operator definition, we developed an algorithm sketch for our CTL model revision approach.

The complete study of semantic properties demands the analysis of whether all constructions that obey the postulates (R1)-(R6) can be represented as our definition of CTL model revision. This will ensure the adequate rationality of our revision operation. Another issue is the analysis of the computational properties of CTL model revision, which is essential before we can think of practical applications.

Acknowledgments. The first author was supported by the Brazilian National Research Council (CNPq). The second author was partially supported by CNPq. This work is part of FAPESP project LogProb, grant number 2008/03995-5.

References

1. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languages and Systems (TOPLAS) 8(2), 244–263 (1986)
2. Amla, N., Du, X., Kuehlmann, A., Kurshan, R.P., McMillan, K.L.: An analysis of SAT-based model checking techniques in an industrial environment. In: Borrione, D., Paul, W. (eds.) CHARME 2005. LNCS, vol. 3725, pp. 254–268. Springer, Heidelberg (2005)
3. Chauhan, P., Clarke, E.M., Kukula, J.H., Sapra, S., Veith, H., Wang, D.: Automated abstraction refinement for model checking large state spaces using sat based conflict analysis. In: Aagaard, M.D., O'Leary, J.W. (eds.) FMCAD 2002. LNCS, vol. 2517, pp. 33–51. Springer, Heidelberg (2002)
4. Cimatti, A., Clarke, E.M., Giunchiglia, F., Roveri, M.: NuSMV: A new symbolic model verifier. In: Halbwachs, N., Peled, D.A. (eds.) CAV 1999. LNCS, vol. 1633, pp. 495–499. Springer, Heidelberg (1999)
5. Buccafurri, F., Eiter, T., Gottlob, G., Leone, N.: Enhancing model checking in verification by AI techniques. Artificial Intelligence 112(1-2), 57–104 (1999)
6. Zhang, Y., Ding, Y.: CTL model update for system modifications. Journal of Artificial Intelligence Research 31(1), 113–155 (2008)
7. Herzig, A., Rifi, O.: Propositional belief base update and minimal change. Artificial Intelligence 115(1), 107–138 (1999)
8. Winslett, M.: Reasoning about action using a possible models approach. In: Proc. of AAAI, pp. 89–93. Morgan Kaufmann, San Francisco (1988)
9. Katsuno, H., Mendelzon, A.O.: On the difference between updating a knowledge base and revising it. In: Proc. of KR, pp. 387–395. Morgan Kaufmann, San Francisco (1991)
10. Alchourron, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. J. Symb. Logic 50(2), 510–530 (1985)
11. Clarke, E.M., Grumberg, O., Peled, D.A.: Model checking. Springer, Heidelberg (1999)
12. Huth, M., Ryan, M.: Logic in Computer Science: Modelling and reasoning about systems. Cambridge University Press, Cambridge (2004)
13. Katsuno, H., Mendelzon, A.O.: A unified view of propositional knowledge base updates. In: Proc. of IJCAI 1989, pp. 1413–1419. Morgan Kaufmann, San Francisco (1989)
14. Herzig, A.: Logics for belief base updating. In: Dubois, D., Prade, H. (eds.) Handbook of Defeasible Reasoning and Uncertainty Management. Vol. Belief Change, pp. 189–231. Kluwer Academic, Dordrecht (1998)
15. Winslett, M.: Updating logical databases. Cambridge University Press, Cambridge (1990)
16. Sousa, T.C.: Revisão de modelos formais de sistemas de estados finitos. Master's thesis, Universidade de São Paulo (2007)
17. Sousa, T.C., Wassermann, R.: Handling inconsistencies in CTL model-checking using belief revision. In: Proc. of the Brazilian Symposium on Formal Methods (2007)

Applying H^m Heuristics in Petri Nets Reachability Problem

Rene Kultz, Luis Allan Künzle, and Fabiano Silva

Tel.: +55-41-8892-1734
renekultz@yahoo.com.br,
kunzle@inf.ufpr.br,
fabiano@inf.ufpr.br

Abstract. The problem of reachability in Petri nets has several approaches. The technique that produces better results is known as "unfolding". However, the size of the unfolded net can be exponential with respect to the initial Petri net. This work aims to adapt planning heuristics to guide the construction of the unfolding. The article analyzes the use of heuristics, based on a regression of the goal state, in a progressive planner. Several experimental results were generated. They are analyzed and compared with those obtained with other planners already established.

Keywords: Planning; Heuristics; Reachability problem; Petri Nets.

1 Introduction

In [5], was proposed a technique to convert planning problems in Petri Nets reachability problems. This work was optimized in [13], which use the concept of multivaluated variables to decrease the number of places and transitions from the generated Petri net, without lose of information. Once the Petri Net is generated, it is unfolded, to obtain the optimal plan from the occurrence net.

The objective of using the unfolding technique is not to obtain the complete prefix, but to obtain a part of the occurrence net that contains the solution. Although the unfolding algorithm generates a net exponentially smaller than the reachability graph, it is exponecially bigger than the initial Petri net from which the occurrence net was generated. To optimize the search, [5] use the H^1 heuristic [4], which is a planning heuristic from the H^m family. [13] used a different heuristic, based on the H^1 heuristic, that is not admissible. This heuristic is based on labeling with integers the places of the Petri net, indicating the smaller number of transitions firings needed to reach any place that belong to the goal state. The heuristic value assigned to a mark is obtained by adding the labels of all places belonging to the marking.

In this article, we propose to use the H^m family to solve the reachability problem. The solution obtained extends the notion of reachability, since we don't treat this problem as a decision problem, i.e., a problem where the algorithm just answer if the goal mark is reachable from the initial mark or not. Besides

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 163–173, 2010.

that, we seek also the sequence of firing transitions that lead from initial state to goal state. The unfolding was done using the tool *Mole* [12], which realize the unfolding from Petri Nets. This tool was been modified to include the H^m heuristics H^1 and H^2 and to use the search strategy A*. Further information about this work can be found in [7].

This algorithm is complete, since the unfolding algorithm explores all reachable markings from the initial one in the Petri Net. If the goal marking is not reachable from the initial one, the unfolding will return the complete prefix and no solution.

2 Planning

Planning problems are a category of problems where the initial state and the goal state are known, and the solution is a sequence of actions that will achieve a goal [11]. In this article, we will consider only classical planning problems, which are fully observable, deterministic, finite, static and discrete.

Let $L = \{p_1, \ldots, p_n\}$ be a finite set of *proposition symbols*. A *set-theoretic planning domain* on L is a restricted state-transition system $\sum = (S, A, \gamma)$ such that [3]:

- $S \subseteq 2^L$, i.e., each state s is a subset of L. If $p \in s$, then p holds in the state of the world represented by s.
- Each action $a \in A$ is a triple of subsets of L, which we will write as $a = (pre(a), add(a), del(a))$. The set $pre(a)$ is called the preconditions of a. The sets $add(a)$ and $del(a)$ are, respectively, the positive efect and the negative efect of a in the state. We require these two sets of effects to be disjoint, i.e., $del(a) \cap add(a) = \emptyset$.
- The state-transition function is $\gamma(s,a)= ((s \ del(a)) \cup add(a))$ if a is applicable to s, i.e., $pre(a) \subseteq s$.

A *set-theoretic planning problem* is a triple $\rho = (\sum, s_0, g)$, where [3]:

- s_0, the initial state, is a member of S.
- $g \subseteq L$ is a set of propositions called *goal propositions* that give the requirements that a state must satisfy in order to be a goal state. The set of goal states is $S_g = \{s \in S \mid g \in s\}$.

The H^m family heuristics [4] are based on computing admissible estimates of minimal path costs of achieving sets of atoms from the initial state s_0. In the H^m algorithm, the state variable sets are divided in subsets of size m. These subsets are regressed by a recursive algorithm, until the final state is reached. The the structure of the problem is static, because the sets of actions and state variables don't change during the search. When the size of the set α is bigger than the specified size m, the set is divided in as many subsets from size m as it is possible, as shown in the item (3) from figure 1. When the size is equals or smaller than m, the value is obtained by the smaller value of all the possible regressions of α

$$H^m(\alpha) = \begin{cases} 0, & \alpha \subseteq s_0 \quad (1) \\ \min\limits_{<\beta,a>\in R(\alpha)} [c(a) + H^m(\beta)], & |\alpha| <= m, \quad \alpha \not\subseteq s_0 \quad (2) \\ \max\limits_{\beta\subset\alpha, |\beta|=m} H^m(\beta), & |\alpha| > m, \quad \alpha \not\subseteq s_0 \quad (3) \end{cases}$$

Fig. 1. H^m heuristic family

from the actions $a \in A$. The process of a regression, represented in the item 2 by the symbol $R(\alpha)$, consists in enumerate all the pairs, $\langle\beta$,a$\rangle$, where $a \in A$, $\alpha \cap add(a) \neq \emptyset$, $\alpha \cap del(a) = \emptyset$ and $\beta = \alpha$ - $add(a) + pre(a)$.

The H^m heuristic family is composed by admissible and monotonic heuristics. The order of H^m heuristics determines both the complexity of calculating as the approximation from the optimal plan value. The complexity of calculating is in the order of $O(n^m)$, where n is the number of state variables and m is the order of heuristic. The higher the value of m, the higher the approximation from the obtained value and the cost to obtain the optimal plan, but it is slower to calculate it. In this article, we will implement H^1 and H^2 heuristics.

3 Petri Nets and Unfolding

A Petri Net [10] is a quadruple $\langle P,T,F,M_0\rangle$, where P and T are disjoint finite sets of places and transitions, respectively. $F : (P \times T) \cup (T \times P) \rightarrow \{1,0\}$ is a flow relation indicating the presence or absence of arcs. A marking M of the net is a mapping $M : P \rightarrow \mathbb{N}$. M_0 is the initial marking. The behavior of a Petri net can be described by a particular labeled net called *occurrence net*.

An occurrence net is an unmarked net (B,E,F',h) where B and E are called conditions and events, respectively, such that: $\forall b \in B, |{}^\bullet b| \leq 1$; F' is aciclic; (B,E,F') is finitely preceded, i.e., for every $x \in B \cup E$, the set of elements $y \in B \cup E$ such that (y,x) belongs to the transitive closure of F' is finite; and h is a labeled function from $B \cup E$ to $P \cup T$. In a occurrence net, conditions without input events are the initial conditions of the net. Each pair $(x,y)|x,y \in B \cup E$ belongs to one of the following conditions: a causuality relation ($x \leq y$ or $x \geq y$); a conflict relation ($x \# y$, where two distinct events have a common precondition); and a concurrency relation ($x \parallel y$, where x and y are neither in causal nor conflict relation).

An unfolding of a 1-safe is an occurrence net which is acyclic, finite and preservs a representation of all reachable markings [9]. This technique was created to avoid the state space explosion problem in reachability analysis of Petri nets. The construction of an unfolding begins with the initial conditions. Transitions enabled at the original net are added to the occurrence net as labeled events, as well as their associated output conditions. This process ends when a particular sets of events called *cutoff* and their descendants are removed from the occurrence net. A complete description of this technique can be founded in [2].

4 Adjusting H^m Planning Heuristics in the Context of Petri Nets Formalism

The H^m vector calculus is a data structure created to help to calculate the heuristics, that enumerate all the sets of size m and its regressings. This structure is used to help the heuristic calculating through the algorithm shown in section 5.1. The creation of the vector calculus is made in 3 steps. First, occours the transcription of the Petri net's transitions. In the second step, we enumerate all the subsets with size smaller or equal to m. In this article, we will threat only the heuristics H^1 and H^2. The third step is enumerate, to each subset, the dependency set, which stores the regressions of the subset.

Be $\mathtt{t} \in \mathtt{T}$ the transition set, the transcription of Petri Nets transitions in STRIPS form are defined by the set $\mathtt{a}_t \in \mathtt{A}$, where, $\forall\ \mathtt{t} \in \mathtt{T}$:

- $\mathrm{pre}(\mathtt{a}_t) = {}^\bullet\mathtt{t}$
- $\mathrm{add}(\mathtt{a}_t) = \mathtt{t}^\bullet \quad ({}^\bullet\mathtt{t} \cap \mathtt{t}^\bullet)$
- $\mathrm{del}(\mathtt{a}_t) = {}^\bullet\mathtt{t} \quad ({}^\bullet\mathtt{t} \cap \mathtt{t}^\bullet)$

After the construction of the transcribed transition, we will enumerate all the subsets of places with size smaller or equals to m. The enumeration is done according the heuristic order. In the H^1 heuristic, it is created one position in the vector calculus to each Petri net's place. In the other hand, the construction of the H^2 is done listing all the pairs of places $(\mathtt{p}, \mathtt{q})$, where $\mathtt{p} \leq \mathtt{q}$. Where $\mathtt{p} = \mathtt{q}$, we consider the regression of sets with size 1. For each position from the vector, it is constructed a structure called *dependencies*, which stores all the possible regressions from the set $\mathtt{p}, \mathtt{q}$.

In the H^1 heuristic, the regression of a set α equals to the *pre* set of the transcripted transition. Be $\mathtt{p} \subseteq \mathtt{P}$ the set of places, $\mathtt{t} \subseteq \mathtt{T}$ the set of transitions and $\mathtt{a}_t \subseteq \mathtt{A}$ the set of transcripted transitions, the vector calculus H^1 is composed by the elements $\mathtt{z}_p \in \mathtt{Z}$. For each $\mathtt{z}_p$, we have a dependency set $\mathtt{c} \subseteq \mathtt{C}_z$, where $\forall\ \mathtt{a}_t \in \mathtt{A} \mid \mathtt{p} \in \mathtt{add}(\mathtt{a}_t), \mathtt{pre}(\mathtt{a}_t) \subseteq \mathtt{C}_z$.

The H^2 heuristic is constructed to obtain the cost to obtain 2 state variables at the same time, and not only one of them. Be $\mathtt{p}, \mathtt{q} \subseteq \mathtt{P}$ the set of places, $\mathtt{t} \subseteq \mathtt{T}$ the set of transitions and $\mathtt{a}_t \subseteq \mathtt{A}$ the set of transcripted transitions, the vector calculus H^2 is composed by the elements $\mathtt{z}_{p,q} \in \mathtt{Z}$. For each $\mathtt{z}_{p,q}$, we have a dependency set $\mathtt{c} \subseteq \mathtt{C}_z$, where:

- $\forall\ \mathtt{a}_t \in \mathtt{A} \mid \mathtt{p}, \mathtt{q} \in \mathtt{add}(\mathtt{a}_t) \Rightarrow \mathtt{C}_z \supseteq \mathtt{pre}(\mathtt{a}_t)$
- $\forall\ \mathtt{a}_t \in \mathtt{A} \mid \mathtt{p} \in \mathtt{add}(\mathtt{a}_t), \mathtt{q} \notin \mathtt{add}(\mathtt{a}_t), \mathtt{q} \notin \mathtt{del}(\mathtt{a}_t)$
  $\Rightarrow \mathtt{C}_z \supseteq \mathtt{pre}(\mathtt{a}_t) \cup \{\mathtt{q}\}$
- $\forall\ \mathtt{a}_t \in \mathtt{A} \mid \mathtt{q} \in \mathtt{add}(\mathtt{a}_t), \mathtt{p} \notin \mathtt{add}(\mathtt{a}_t), \mathtt{p} \notin \mathtt{del}(\mathtt{a}_t)$
  $\Rightarrow \mathtt{C}_z \supseteq \mathtt{pre}(\mathtt{a}_t) \cup \{\mathtt{p}\}$

To optimize the calculation, all the value of the *pre* sets can be stored in another structure, to avoid to calculate the same value many times. When a value is defined, it can be used for all the others dependencies which uses this value. As the dependencies sets are constructed storing the index of the sets found in the

vector calculus, when the vector calculus is constructed and when the heuristic is calculed it is necessary to know the index of the sets. It is very expensive, considering time, to search in the H^2 vector, the index of the sets. Due this fact, we developed formulas, based in the H^2 vector structure, which function is to calculate the H^2 vector size and to find any valid set `p,q`, showed in the figure 2.

$$S_n = \frac{n * (1+n)}{2}$$

(a) Size of the vector H^2

$$index(p,q)_n = \frac{p * (n + (n - p - 1))}{2} + (q - p)$$

(c) Index of the vector H^2

Fig. 2. H^2 vector's formulas

5 Solving the Reachability Problem

In this section, we show how to use the vector calculus to calculate the H^1 and H^2 heuristics, and how to use these informations to guide the unfolding process to solv the reachability problem.

5.1 Calculation of the Heuristic

The heuristic is calculated through the vector calculus. In the first place, the vector calculus is constructed, according the process shown in the section 4. To calculate the heuristic, we use the following algorithm.

The vector's inicialization consists in, first, fill all the vector calculus heuristics with ∞ value, represented in the code by a high value. In the second place, all the vector calculus heuristcs where all the subsets elements belongs to the marking m are filled with value 0. After the inicialization, the calculus can be done associating to each vector calculus's subset, a value, according the value found in the dependency set. The value is defined extracting the bigger value in each dependency and the smaller value found along the dependencies. The algorithm is executed until the point where all the vector calculus's heuristics are filled until the point when, after a loop, no one heuristic value is updated.

5.2 Optimizations Resulting from the Initial Marking

The initial marking influences directly the reachability problem in Petri nets. To obtain some informations about the state space, we fill the vector calculus using the initial marking $\mathbb{m}_0$. After this, all the vector calculus positions which heuristics values are equals to ∞ represents subsets that are not reachable from the initial state. Consequently, no states that contains the subset presents in the vector calculus positions are reachable from the initial state, and these positions can be eliminated from the vector calculus. Similarly, all the dependencies that contains, at least, one element that is not reachable from the inicial state, can be eliminated.

```
Input: vector calculus Z, marking m, marking goal
Output: max value found in goal positions of Z
initialize the vector calculus;
flag ← TRUE;
while flag do
    flag → FALSE;
    foreach vector calculus position z_i ∈ Z do
        if z_i.heuristic = ∞ then
            min_value ← ∞;
            foreach vector calculus's dependency c_j ∈ C_i do
                max_value ← max value found in c_j;
                if (max_value ≠ ∞) AND (max_value + 1 < min_value)
                then
                    min_value ← max_value + 1;
                end
            end
            if min_value ≠ ∞ then
                flag ← TRUE;
                z_i.heuristic ← min_value;
            end
        end
    end
end
```

Algorithm 1. Algorithm to obtain an heuristic value from the vector calculus

All the remaining vector calculus's positions are ordered according the heuristic values obtained by the initial marking, because the order generated by the construction of the vector calculus is related to the position that the places are in the input file from Mole. Although this order will chance during the seach, it is better than the old order.

5.3 Search for the Optimal Plan

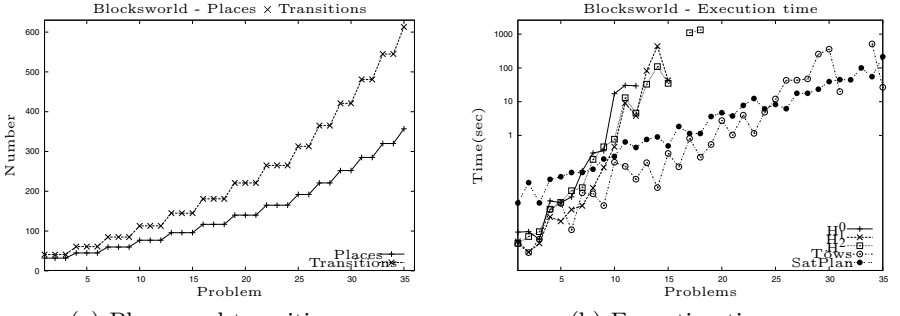
The tool Mole was modified to implements the search strategy A*. In the original version, the priority list from Mole who stores the non-expanded events is ordered by the size from the local configuration from the event. Due this, it realizes the unfolding according to the logic from the breadth-first search algorithm. In the modified version, the priority list is ordered by the function `f(e) = g(e) + h(e)`, where `g(e)` is the size from the local configuration from the event `e`, representing the optimal cost to obtain this event, and the function `h(e)` represents the heuristic function applied on the marking associated to the event `e`. The search is done until Mole tries to expand the event who contains the stop transition. Then, Mole will stop the unfolding process and the optimal plan can be extracted from the occurrence net. The optimal plan corresponds to the local configuration from the event who contains the stop transitions.

6 Experimental Results

In this section, we will show experimental results obtained from Petri nets generated from planning problems using the algorithm *Graphplan*, proposed by [13]. The domains where extracted from the planning competitions AIPS-2000[1] e IPC-2002[8]. All the experiments were conducted on a 2 Opteron Dual Core, with 32 Gb of memory, using Debian Linux. These experiments were submitted with a time limit of 2 500 seconds. The problems where this version of Mole found a solution in a time smaller than this will be analyzed according to the number of expansions. We will compare the time used by the heuristics H^1 and H^2 with the time used by the tool SatPlan [6] and with the time used by the heuristic implemented by Töws [13]. In the problems where the solution was not found in the given time, we will analyze how closed to the solution the implemented heuristics was gone. The notation H^0 refers to execution from Mole without use of heuristics to guide the unfolding process.

6.1 Blocksworld

The domain Blocksworld describes a family of problems based on operations from a set of blocks in a table. The figure 3(a) shows the number of places and transitions presents in the Petri nets generated from these problems.



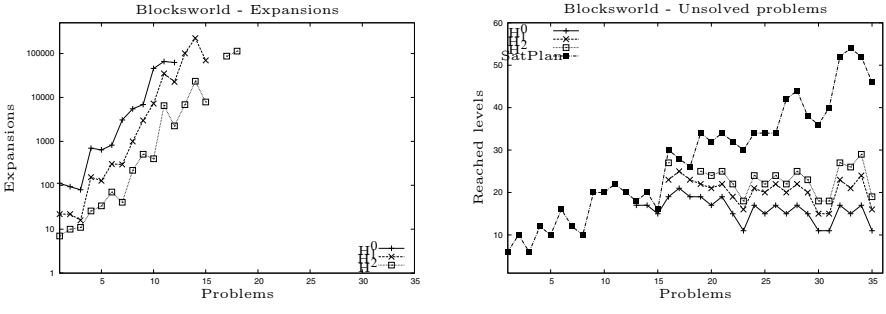
(a) Places and transitions (b) Execution time

Fig. 3. Blockswold

The Petri nets represented from the number 1 to 9, which biggest net have 60 places and 85 transitions, have their optimal plan found in a time smaller than 1 second with any heurstics. In the Petri nets represented by numbers biggest than 13, the execution time increase significantly. In the Petri nets 13, 14 and 15, the heuristic H^0 didn't find the optimal plan in less time than 2 500 seconds. In the problems 16 and 18, only the heuristic H^2 found a solution in a time smaller than the time limit. The execution time from the search oriented by the heuristic implemented by Töws was smaller than the execution time from the

H^1 and H^2 heuristics, in all the cases. The reason is that the Töws's heuristic is not admissible, and the own search is not based on the A* algorithm, but on the greedy search algorithm. Comparing with the SatPlan, the search was more efficient in the small problems, where the time was smaller than 1 second, but the SatPlan became more efficient as the problems's complexity increases.

The number of expansions realized during the search is shown in the figure 4(a). In all the cases, the H^2 found the optimal plan with less expansions than the expansions necessary by the heuristics H^1 and H^0. The number of expansions is directly related to the difference of time between the heuristics. To realize a new expansion, the unfolding algorithm needs to search in the occurrence net by enabled conditions, i.e., conditions that are not preconditions from any event. This search becomes more slower as the size of occurrence net increases. The H^2 heuristic is much more slower to cal than the H^1 heuristic, because the complexity of the H^2 vector is in the order of $\mathtt{n}^2/2$, considering $\mathtt{n}$ as the number of places in the original Petri net. In the smaller Petri nets, as the Petri nets from 1 to 13, the unfolding algorithm is too fast. Because of this, the solution is found by H^1 and H^0 heuristics before the H^2 heuristic, although the H^2 heuristic have been realized less expansions. In the medium problems, the occurrence net's size becomes to make the new expansions more slowly, and the difference of complexity from the H^2 heuristic is compensated by the quick expansion from the occurrence net.



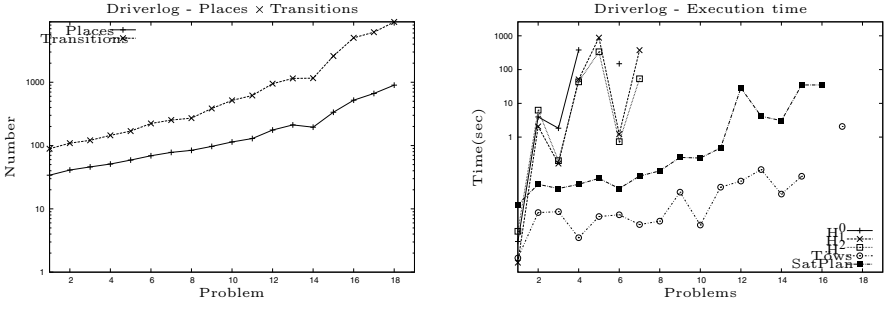
(a) Expansions (b) Unsolved problems

Fig. 4. Blockswold

In the biggest problems, the complexity from the vector calculus and the own number of expansions increases too much, due the problem complexity, and no one heuristic becomes efficient for solving the problems in less than 2 500 seconds. The figure 4(b) show the depth from the unsolved problems. As the problems's complexity increases, the efficiency from the heuristics decreases, but in all cases the H^2 heuristic reached a position more closed to solution in the occurrence net than what was found in the H^1 or H^0 heuristics.

6.2 Driverlog

The domain Driverlog describes a logistic problem, where packages need to be transported by trucks. The figure 5(a) shows the number of places and transitions from the Petri nets.



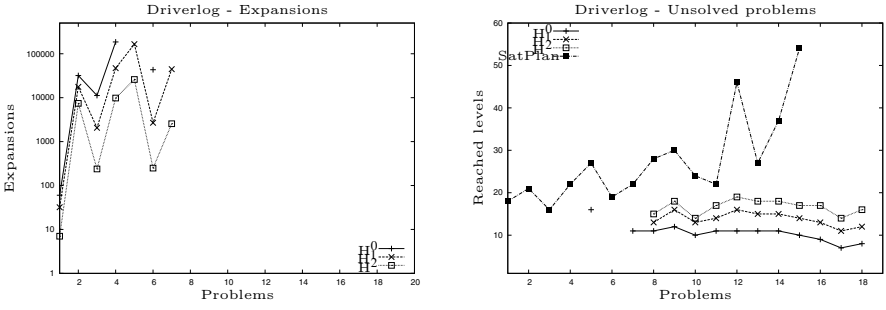
(a) Places and transitions (b) Execution time

Fig. 5. Driverlog

The Petri nets generated from the Driverlog domain are more complex than the Petri nets generated from the Blocksworld domain. While, in the Blocksworkd, the number of places varies between 32 and 613 and the number of transitions varies between 42 and 613, in the Driverlog domain the number of places varies between 34 and 900 and the number of transitions varies between 89 and 8961.

According to figure 5(b), only on the first example the execution time was smaller than 1 second for all the heuristics. In the problems 3 and 6, only the H^1 and H^2 heuristics found the optimal plan in less time than 1 second. The driverlog problems tends to generate more events when an expansion is made, what makes the own search more slowly when compared to the search made by the SatPlan. The Töws search returns the first solution found, making the test when the event is created. In the case of H^m heuristics, an event is only considered a solution when it is in the first position from the priority list.

The number of expansions is shown in the figure 6(a). The increase from the complexity from the problems reflected to an increase from the number of expansions. In all the cases, the H^2 heuristic realized less expansions than the number of expansions from the H^1 and H^2 heuristics. The increase from the complexity reflected in the depth reached by the unsolved problems, which is shown in the figure 6(b). Even the SatPlan didn't found a solution for all the problems. In the same way as was found in Blocksworld, the H^2 heuristic got a level more close to solution than the other heuristics, but the difference between them are smaller.



(a) Expansions (b) Unsolved problems

Fig. 6. Driverlog

7 Conclusion

In this article, we tried to adapt the planning heuristics H^m to the tool Mole, modifying the unfolding algorithm to work as the logic from the search algorithm A* for solving reachability problems in Petri nets. To reach this objective, we studied the H^m heuristics and manners to adapt them in the Petri nets formalism. It was done by a data structure called "*vector calculus*", which function is enumerate all the subsets and their respective regressions. The data structure allowed that the regressions form the sets was maid as pre-processing, and allowed some optimizations resulting from the initial marking. It makes the calculation of the heuristic faster.

The experiments showed that the H^2 heuristic was less efficient when compared to the H^1 heuristic in smaller problems, which execution time is smaller than 1 second. As the problem complexity increases, the efficiency from the H^2 increases when compared to H^1 and H^0 heuristics, as long the number of required expansions increases. In the biggest problems, none of heuristics was efficient, because the complexity from the heuristic calculus and the time spent to expand the occurrence net increase. Although, the use of H^2 heuristic makes the unfolding algorithm arrive closer to the solution than the H^1 or H^2 heuristics.

The experiments was realized using only the H^1 and H^2 heuristics, but the data structure used permits to use another heuristics from order bigger than 2. As future works, we pretend to use another heuristics, like H^3.

References

1. Bacchus, F.: The aips 2000 planning competition. AI Magazine 22(3), 47–56 (2001)
2. Esparza, J., Römer, S., Vogler, W.: An improvement of mcmillan's unfolding algorithm. In: Formal Methods in System Design, pp. 87–106. Springer, Heidelberg (1996)
3. Ghallab, M., Nau, D., Traverso, P.: Automated Planning - theory and practice, 1st edn. Morgan Kaufmann, San Francisco (2004)

4. Haslum, P., Geffner, H.: Admissible heuristics for optimal planning. In: Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems, pp. 140–149. AAAI Press, Menlo Park (2000)
5. Hickmott, S., Rintanen, J., Thi'ebaux, S., White, L.: Planning via petri net unfolding. Technical report. In: ECAI-Workshop on Model Checking and Artificial Intelligence (2006)
6. Kaunts, H., Selman, B.: Planning as satisfiability. In: Proceedings of the 10th European Conference of Artificial Inteligente (ECAI 1992), pp. 359–363 (1992)
7. Kultz, R., Künzle, L.A.: Utilização de heurísticas de planejamento no desdobramento de redes de petri. Master's thesis, Universidade Federal do Paraná (2010)
8. Long, D., Fox, M.: The 3rd international planning competition: Results and analysis. J. Artif. Intell. Res. (JAIR) 20, 1–59 (2003)
9. Mcmillan, K.L., Probst, D.K.: A technique of state space search based on unfolding. In: Formal Methods in System Design, pp. 45–65 (1992)
10. Reisig, W.: Petri Nets - An Introduction. Springer, Heidelberg (1985)
11. Russell, S., Norvig, P.: Artificial Intelligence - A Modern Approach, 2nd edn. Prentice Hall, Englewood Cliffs (2003)
12. Schwoon, S., Römer, S.: Mole - an unfolder for low-level petri nets (2004), Free software under GPL licence, `http://www.fmi.uni-stuttgart.de/szs/tools/mole/mole-060323.tar.gz`
13. Töws, G.S., Silva, F.: Petrigraph: um algoritmo para planejamento por desdobramento de redes de petri. Master's thesis, Universidade Federal do Paraná (2008)

Sampling for Information and Structure Preservation When Mining Large Data Bases

Angel Kuri-Morales[1] and Alexis Lozano[2]

[1] Departamento de Computación,
Instituto Tecnológico Autónomo de México,
Rio Hondo No. 1,
Mexico City, Mexico
[2] Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas,
Universidad Nacional Autónoma de México
Ciudad Universitaria,
Mexico City, Mexico
akuri@itam.mx, lozano_a@uxmcc2.iimas.unam.mx

Abstract. The unsupervised learning process of identifying data clusters on large databases, in common use nowadays, requires an extremely costly computational effort. The analysis of a large volume of data makes it impossible to handle it in the computer's main storage. In this paper we propose a methodology (henceforth referred to as "FDM" for *fast data mining*) to determine the optimal sample from a database according to the relevant information on the data, based on concepts drawn from the statistical theory of communication and L_∞ approximation theory. The methodology achieves significant data reduction on real databases and yields equivalent cluster models as those resulting from the original database. Data reduction is accomplished through the determination of the adequate number of instances required to preserve the information present in the population. Then, special effort is put in the validation of the obtained sample distribution through the application of classical statistical non parametrical tests and other tests based on the minimization of the approximation error of polynomial models.

Keywords: Large databases, Sampling, Data mining, Clustering, Preprocessing.

1 Introduction

In the last decades the increase of information flow and the resulting need of storage have grown at extremely fast rates due to the availability of powerful computers, large storage devices and automatic data acquisition tools. Even though all this large volume of data is accessible and useful, it poses several challenges. These problems span the methods and hardware specific difficulties to access and browse it, on the one hand, to the complexity of extracting some significance, on the other.

The storage of this data is generally done on databases, where the data is structured and organized to ensure future accesses and preserve the information. Large databases used on enterprises gave rise to data warehousing: the use of are large raw data

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 174–183, 2010.

repositories and tools designed to facilitate the data analysis and reports, generally including business intelligence tools (consisting on the application and interpretation of the data mining techniques). This is a key element on medium to large sized organizations as a decision support tool [1].

We will define data mining as the set of techniques to obtain (automatically or semi-automatically) interesting patterns from large volumes of data. It is an important area of research because of its practical benefits and the necessity of its results in many commercial and academic fields. However, the problems that these techniques have to deal with are highly complex and depend on several factors. For instance, the efficiency of the algorithms, the data representation or the quality of the data. The quality of the information is hard to measure. In practice, it is difficult to exactly determine the most important characteristics from a population given a certain aim; this would imply solving great part of the data mining problem. Therefore, no absolute methodology has been set upon the process of selecting the correct data for a data mining processes.

The problems associated with the last issue have been analyzed from different perspectives [2]: optimizing the process through the use of parallel computation, optimization of algorithms with different approaches (such as divide and conquer techniques [3] [4]), sequential algorithms [5], or feature selection [6]. The main focus of this paper, however, is on the reduction of data dimensionality to obtain a representative sample that importantly reduces the cost of the process with similar results to those obtainable from the population itself.

In data mining data reduction is crucial. It can be achieved if a subset of the analyzed data contains enough information through a representative sample. Sampling, therefore, is fundamental in the analysis of data. Most of the efforts have focused on a vertical reduction through correlation [7] or principal components analysis [8]. Some other researches have centered on instance selection or a sampling method in which the distribution of the data is not assumed a priori. Most of the existing works deal with the efficiency of the sampling itself [9], or the sample determination in one dimension. This paper focuses on the definition of a methodology to extract a representative sample taking advantage of the fact that the population is owned[1] by the user. We further emphasize the preservation of the relations and structure of the data. The results are validated with different methods which are shown not to share some of the limitations of traditional statistics while, at the same time, preserving low computational costs.

2 Method FDM

2.1 Problem Definition

Data mining techniques allow for the analysis of large volumes of data in order to extract meaningful information. It comprises a set of techniques that explore the data with the aim of obtaining models, patterns or rules to compactly and more meaningfully describe the data set thus enabling us to achieve reliable predictions. These techniques usually borrow from artificial intelligence, statistics and/or databases management.

[1] In the social or biological sciences, in contrast, the population is not usually known ("owned") and the researcher has to settle with coarser estimates.

A fundamental set of data mining algorithms are the clustering techniques. Clusters give a first insight to the data structure allowing the application of other processes to model or to achieve finer samples. Also, important inference can be derived from the classification results, i.e. predictive models, the effect of certain experimental applications, or functional associations. Clustering consists on grouping the data into sets that hold similar characteristics within and exhibit clear differences without. This is a non trivial process because the similarity measures are not always clear, and because these similarities should be validated between all possible subsets and subspaces of the data. In general, the search for data cluster is an NP complete combinatorial optimization problem [10]. Therefore, artificial intelligence and statistical approaches are commonly used to solve it. For a reference of clustering algorithms see [11].

Any clustering algorithm be it partitional, hierarchical or AI-based[2] will always depend on the distribution of the data. Hence, any data economization process should generate a sample that preserves the variability and structure of the whole data set.

It is important to remark that the inherent statistical vagueness of any sampling process is not a disadvantage. Rather, it will allow us to view the system under a more general perspective. That is, the model is not required to be exact over the data set in order to keep its ability to generalize. This justifies the use of economization techniques without significantly modifying the obtainable model.

2.2 FDM Description

2.2.1 Data Preparation

FDM (as most others alternatives) requires data preprocessing targeted at avoiding predictable misbehavior of the data.

The first step is the preparation of the data from the database which generally is distributed on normalized tables. It should be joined and processed in order to obtain a simple view to the data of interest in the form of a set of attributes on a single table.

The data may contain defects that may hamper the analysis performance such as missing values or inconsistent data. For a guideline of the steps needed to handle these problems see [12]. Also, the type of the variables is vital. Most of the techniques (such as the ones discussed in this paper) numerically analyze the data. Therefore, non numerical variables must be very carefully encoded or plainly eliminated.

It is also convenient to normalize the data to reduce the probability of numerical instability while processing the data and to balance the weight of the variability on each dimension of the data. These preprocessing stages may result in some loss of information. The best way to tackle its shortcomings is still an open area of research. Furthermore, we shall assume that the data is representative of the system under study.

2.2.2 Sample Size Determination

We discuss a methodology to select a representative sample for a clustering mining process. The data set is analyzed as a set of sequences of symbols composing messages in accordance with the statistical theory of communication. Each set of attributes represents a message and each attribute value corresponds to a symbol. In this light is possible to estimate the entropy contained on the message. The entropy is a measure of

[2] For instance, fuzzy c-means or self organizing neural networks.

the average information in the symbols of a message. It is a tool which will be used to ensure that the information of the sample and that of the population are alike.

The entropy of the message can be approximated as the proportion of appearances of each symbol as follows:

$$H(X) = \sum_{i=1}^{m} - p_i \log p_i \approx \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \frac{\delta(S_i, v_i)}{n} \right) \log \left(\sum_{i=1}^{m} \frac{\delta(S_i, v_i)}{n} \right) \quad (1)$$

Where X is the message, p_i is the probability of occurrence of symbol *i*. *m* is the number of symbols, *n* is the number of data elements, S_i is the *i-th* symbol value and v_j is the *j-th* data value and

$$\delta(s, v) = \begin{cases} 0 & if \ v \neq s \\ 1 & if \ v = s \end{cases} \quad (2)$$

Our aim is to approximate the value of the population's entropy by the entropy of a properly selected sample to avoid accessing the full data in the database. The method consists of treating every attribute *t* to obtain a sample M_t, as follows. Initially M_t is empty. Then we proceed to extract uniformly selected elements of the population for attribute *t* iteratively and adding these elements to M_t. On each iteration *i,* the entropy is calculated and compared with the one of the previous iteration as follows

$$\Delta H(i) = H(i) - H(i-1) \quad (3)$$

As ΔH(i) becomes closer to a parameter ε the entropy of M_t is asymptotically closer to the population's entropy as illustrated in figure 1. At this point the size $|M_t|$ has been determined. The value of ε is set by the user according with the validation tests that will be discussed in what follows.

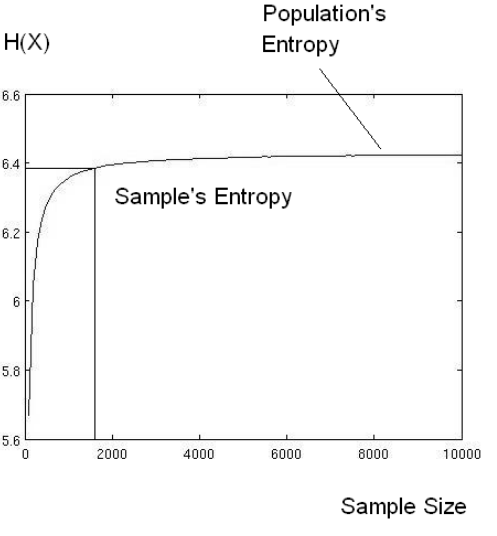


Fig. 1. Entropy dynamics on different sample sizes

This process is performed, as stated, on every attribute. Once the M_i's have been calculated then the overall sample size M is the largest one, thus ensuring the proper representativeness for all attributes.

2.2.3 Sample Validation

To further attest to the validity of any possible sample of size $|M|$, we must ensure that their entropies have the same or larger value than the one calculated before.

Making use of the information available, a simple non parametric Monte Carlo test may be applied [13] to validate the entropy preservation on each variable. This test should confirm the null hypothesis H_0: $H(X) \leq H_C(X)$ where $H(X)$ is the value of the maximum entropy obtained from the sample size calculation process and $H_c(X)$ is the calculated entropy of any sample of size $|M|$.

The test consists of the generation of a set of samples of size $|M|$. If we set a confidence level of 95%, then at least 95% of the samples must be consistent with the null hypothesis, proving that the samples will comply. If the test is not passed then the sample size should be increased.

Because of computational cost, we only perform tests between couples of attributes. In strict sense, to validate the preservation of the all patterns and relations, multivariate tests of high order should explored. Nevertheless, a mathematical model may be calculated to estimate the value of an arbitrary attribute as a function of another, also arbitrarily selected. We calculate a minimax polynomial for reasons that will be discussed later. Our thesis is that validation between couples will suffice in most cases. Then we proceed as follows.

Once a minimax polynomial is obtained from a sample then it can be evaluated on data from different samples. We calculate the approximation L_2 error for polynomial P, where $P(X_i)=Y_i+e$. Where X_i, Y_i are the values of the *i-th* value of the attributes and e is the L_2 approximation error on that data value. This error is defined as:

$$e = \sqrt{\sum_{i=1}^{n} \left(P(X_i - \dot{Y}_i)^2 \right)}$$

If the sample preserves all the couples' relations then the approximation error should be close on every sample. We calculate the ratio:

$$r = e_{max}/e_{min}$$

r must be close to 1 if the approximation errors are similar. A sample could be rejected if $r > 1+\gamma$ (where γ's value is determined by the user). This analysis should be applied to every couple to determine whether to accept or increase $|M|$.

Once M has been validated we are statistically certain that a clustering algorithm operated on it will yield similar results as if it were applied to the original database.

3 FDM Analysis

3.1 Sampling Scheme

We assume that the sampling acquisition process is efficient and the time to access to any data record is constant. Further, we assume that the samples taken from the

population are uniform probabilistic. This sampling scheme is asymptotically efficient; its only limitation is that it may need larger sample sizes to attain representativeness if there are significant proportionality irregularities in the data. Alternative methods based on the density of the data are systematic sampling [14] [15] and stratified sampling [16] schemes. However, they are not appropriate in our case due to the potentially large number of dimensions.

3.2 Entropic Sampling

The use of the entropy as a measure to preserve the integrity of clusters is possible for the following reasons. a) The capability to identify clusters in the data set is based in the detection of similarities on a set of elements, and these similarities imply the existence of certain patterns. In this sense a pattern is a subset of data characteristics that appear on a subset of the analyzed data. b) The presence of patterns is ubiquitous on every data set that is not totally random, implying that there could be some codification that, taking advantage of the existence of those patterns, can express the message in a more compact way. c) Hence, there is a direct relationship between the existence of patterns and the codification length. In other words, there is a relation between findable clusters and the entropy on the data. d) Consequently, data reduction schemes must preserve the entropy of the variables, so that the optimal codification is the same or very similar to the one of the complete data set. e) Thus, it is a good statistic to guide the sampling process.

Some adaptations may be explored in cases where there are complex data sets. For example, when having very widely dispersed variable, the use of a fixed value as a symbol might not be appropriate. Instead, an interval should be considered a symbol, thus avoiding the possibility of estimating a uniform probability distribution on the data just because of the non significant differences between data values.

The patterns on the population are distributed over multidimensional subspaces, so that, generally, there is less pattern information the smaller a subset of variables is. This suggests that the sample size should be calculated from the joint entropy of the set of all variables. However, for practical reasons, we restrict ourselves to the independent calculation of the entropy of every attribute. It is much faster and holds a direct relationship with the joint entropy and the clusters' characteristics. This is due to the fact that multidimensional relations are preserved if the sample reasonably maintains the distribution of each of the variables. However we stress that it is not possible to reflect multivariate behavior by that of a small subset of the variables. We can only say that the sample with enough information of each variable will have higher probability of preserving the multidimensional patterns, and therefore that multidimensional validation tests are desirable.

3.3 Bivariate Validations

To test the multidimensional relations between variables we tried some multivariate statistical tests. They turned out to be not appropriate because they have been typically used on parametric tests assuming normality. The nonparametric alternative for multivariate test, based on generalizations of the univariate tests (like Kruskal-Wallis and Kolmogorov-Smirnov tests) present several limitations. For instance, the loss of detail or the need to order the data. This last is unfeasible with high

dimensional data because a natural ordering scheme is not possible. For these reasons we postulate an approximation error test as a more reasonable alternative.

We selected to approximate the data points with a polynomial because it has the capability of making a very good approximation of any analytic function [17]. This encompasses a wide range of possible functions. Now, we point out that exact polynomial fitting algorithms (based on the minimization of the sum of quadratic errors) involve the solution of a system of equations of the size of the data set. Therefore, this process is impractical on large data sets like the ones that compose a database. Iterative L_2 algorithms (such as Levenberg-Marquardt), on the other hand, are computationally costly on large data sets. Furthermore, they are not amenable to multivariate generalization. A practical alternative not displaying the said limitations is the Ascent Algorihm (AA) [18] which relies on minimizing the L_∞ norm. Its application yields the so-called minimax polynomial. Its calculation involves the iterative solution of a set of only $n+2$ equations where n is the polynomial's degree. In every case, the approximation test incrementally obtains a minimax polynomial of degrees i=2,…,12. This fact emphasizes the need for a very efficient approximating polynomial. The degree of the best approximant was retained.

4 Tests and Results

To test our method we applied it to representative data; some were automatically generated with interesting distributions in X-Y (2 attributes and 50,000 tuples) and some stemmed from a real world enterprise database with 37 attributes (65,535 tuples; 37D). The generated X-Y data sets are shown on figure 2.

As shown in table 1 $|M|$ is much smaller than the original data set. The artificial data sample complies with $\gamma=0.1$. This is true over the different samples as may be seen in table 2.

The same test of bivariate relations was performed on the 37 dimensions data set. The result complied for the said γ for every variable but one. This reflects the fact that such variable is devoid of information. As a corollary of FDM, potentially problematic variables can be analyzed. In this case the variable had a very low entropy value; only a few data points were dispersed and a large accumulation on a certain value was easily detected. Because of this it is impossible to detect those few disperse values in small samples. Interestingly, this fact did not affect the clustering process that ensues.

We wish to point out that the present methodology has been used, in practical problems, on databases whose size is of the order of several million attributes and more than one hundred variables. Obviously, in a work of this kind, it is impossible to attempt the description of these extensive data sets. Furthermore, it has been successfully tested with the data sets of large enterprises. We are unable to discuss these cases in detail for confidentiality reasons. Suffice it to say, then, that the mentioned data come from various entrepreneurial sources: telephone, banks, pharmaceutical industry and insurance companies.

We now evaluate the performance of the clustering on M. A clustering algorithm is applied to the whole set, and then the same algorithm is applied to M. We used a self organizing neural network which is a rugged clustering alternative. The steps to use the algorithm were as follows.

- The data clusters are determined for the complete data set, and the data is labeled with the number of the cluster which it belongs to.
- Likewise, the clusters on *M* are calculated and the data is labeled.
- The clustering model obtained from *M* is applied to the population to label again the elements with *M* and the same is made on data of *M* using the population model.
- The proportion of elements equally classified both on *M* and on the population is calculated.

In table 3 we see that the classification results are practically equivalent even though the data sets exhibit highly non linear relations.

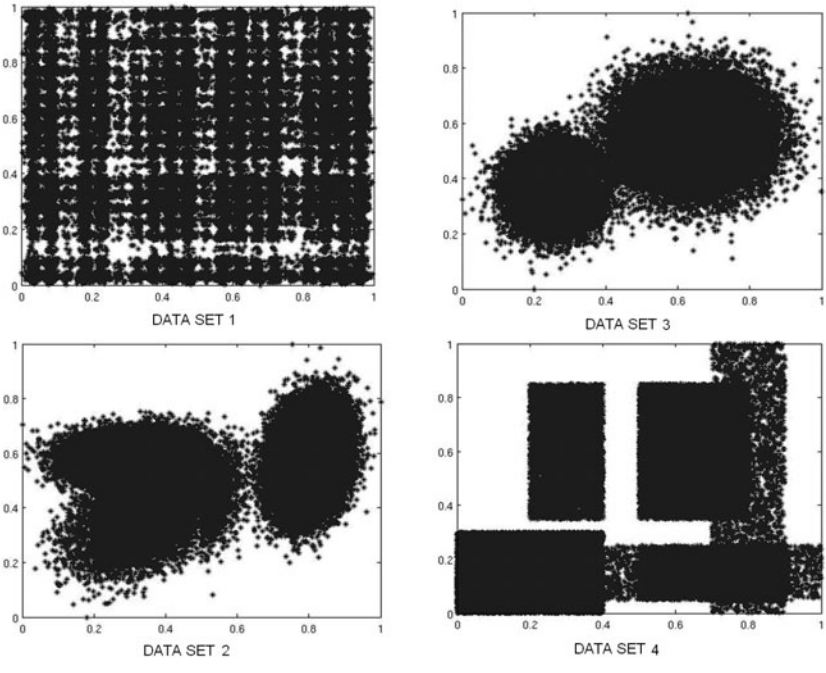


Fig. 2. Plot of the generated data sets

Table 1. Sample size obtained for different data sets

Data set	Original data size	Sample sizes	Efficiency
Data set 1	50000	1500	0.97
Data set 2	50000	3000	0.94
Data set 3	50000	1500	0.97
Data set 4	50000	2000	0.96
37 dimensions data set	65535	9100	0.87

Table 2. Validation results on the artificially generated data sets taking 30 different samples

Data set	Error ratio for P(X1)=X2	Error ratio for P(X2)=X1
Data set 1	1.0762	1.0961
Data set 2	1.0205	1.0902
Data set 3	1.0358	1.0962
Data set 4	1.0657	1.0871

Table 3. Proportion of equally classified data on the data sample and population

Data set	Error ratio for P(X1)=X2	Error ratio for P(X2)=X1
Data set 1	1.0762	1.0961
Data set 2	1.0205	1.0902
Data set 3	1.0358	1.0962
Data set 4	1.0657	1.0871

5 Conclusions and Future Works

FDM achieves a data dimensionality reduction through a sampling process based on the entropy of the data. It allows us to extract samples with equivalent information and structure as the one of the population. This translates in less need for mass storage media and much faster potential exploration of the strategic data. Since the main problems in data mining have to do with both issues, we may obtain important economic benefits from the application of FDM. More importantly, faster (*much* faster) query response times allow us to explore many more potential strategic key issues than before.

The bivariate structural relations were thoroughly tested. The minimax polynomials approximation used in the bivariate relation tests can be easily generalized to any number of dimensions and may offer an alternative to test the results with acceptable costs and yielding useful model interpretations. Since the implemented algorithm may be generalized to, virtually, any number of dimensions, we expect to conduct these experiments and test the possible limitations of bivariate vs. multivariate validation in the very near future.

References

1. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. AI Magazine (1996)
2. Kriegel, H.P., Kroger, P., Zimekm, A.: Clustering High-Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering. ACM Trans. Knowl. Discov. Data 3(1), Article 1 (2009)
3. Cheng, D., Kannan, R., Vempala, S., Wang, G.: A divide-and-merge methodology for clustering. ACM Trans. Database Syst. 31(4), 1499–1525 (2006)

4. Jagadish, H.V., Larkshmanan, L.V., Srivastava, D.: Snakes and Sandwiches: Optimal Clustering Strategies for a Data Warehouse. In: ACM Proceedings: International Conference on Management of Data, Philadelphia, USA, pp. 37–48 (2001)
5. Raïssi, C., Poncelet, P.: Sampling for Sequential Pattern Mining: From Static Databases to Data Streams. In: Seventh IEEE International Conference on Data Mining (2007)
6. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3 (2003)
7. Pitt, E., Nayak, R.: The Use of Various Data Mining and Feature Selection Methods in the Analysis of a Population Survey Dataset. In: Proc. 2nd International Workshop on Integrating Artificial Intelligence and Data Mining (AIDM 2007) (2007)
8. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer Series on Statistics (2002)
9. Olken, R., Rotem, D.: Random Sampling from Databases - A Survey (1995), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.8131
10. Slagle, J.R., Chang, C.L., Heller, S.: A Clustering and data-reorganization algorithm. IEEE Trans. on Systems, Man and Cybernetics (1975)
11. Berkhin, P.: A Survey of Clustering Data Mining Techniques. In: Grouping Multidimensional Data. Springer, Berlin (2006)
12. Zhang, Y., Zhang, J., Ma, J., Wang, Z.: Fault Detection Based on Data Mining Theory. Intelligent Systems and Applications, pp. 1–4 (2009)
13. Zhu, L.: Nonparametric Monte Carlo Tests and Their Applications. Springer Science+Business Media, Inc., Heidelberg (2005)
14. Guo, H., Hou, W.C., Yan, F., Zhu, Q.: A Monte Carlo Sampling Method for Drawing Representative Samples from Large Databases. In: Proceedings of the 16th International Conference on Scientific and Statistical Database Management (2004)
15. Palmer, P., Floutsos, C.: Density Biased Sampling: An Improved Method for Data Mining and Clustering. In: Proceedings of ACM SIGMOD International Conference on Management of Data (2000)
16. Saunders, I.: Restricted stratified random sampling. International Journal of Mineral Processing 25(3-4), 159–166 (1989)
17. Krantz, S., Parks, H.R.: A Primer of Real Analytic Functions, 2nd edn. Birkhäuser, Basel (2002)
18. Cheney, E.W.: Introduction to Approximation Theory. AMS Chelsea Publishing (1998)

Improved Graph-Based Metrics for Clustering High-Dimensional Datasets

Ariel E. Bayá* and Pablo M. Granitto

CIFASIS, French Argentine International Center for Information and Systems Sciences, UPCAM (France) / UNR–CONICET (Argentina),
Bv. 27 de Febrero 210 Bis, 2000 Rosario, Argentina
{baya,granitto}@cifasis-conicet.gov.ar
http://www.cifasis-conicet.gov.ar/

Abstract. Clustering is one of the most used tools for data analysis. Unfortunately, most methods suffer from a lack of performance when dealing with high dimensional spaces. Recently, some works showed evidence that the use of graph-based metrics can moderate this problem. In particular, the Penalized K-Nearest Neighbour Graph metric (PKNNG) showed good results in several situations. In this work we propose two improvements to this metric that makes it suitable for application to very different domains. First, we introduce an appropriate way to manage outliers, a typical problem in graph-based metrics. Then, we propose a simple method to select an optimal value of K, the number of neighbours considered in the k-nn graph. We analyze the proposed modifications using both artificial and real data, finding strong evidence that supports our improvements. Then we compare our new method to other graph based metrics, showing that it achieves a good performance on high dimensional datasets coming from very different domains, including DNA microarrays and face and digits image recognition problems.

1 Introduction

Clustering is a key component of pattern analysis methods, aiming at finding hidden structures in a set of data. It is a very active field of research [1,2,3], with applications that cover diverse problems, from grouping star systems based on astronomical observations to selecting genes and proteins that have similar functionality or characterizing customer groups based on purchasing patterns.

The problem of finding a fixed number of clusters in a dataset can be divided into three stages: i) measuring the similarity of the objects under analysis, ii) grouping the objects according to these similarities, and iii) evaluating the "goodness" of the clustering solution. The second stage (finding clusters efficiently given a set of similarities between objects) has been widely studied in the literature [4] and several clustering algorithms have been introduced. The last stage has received little attention until recent years, when a growing interest in the problem can be noticed [5,6].

* Corresponding author. Work supported by ANPCyT grants 643 and 2226

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 184–193, 2010.

Despite the success that most methods have shown in several artificial and experimental datasets, they usually fail to handle manifold-structured data, i.e. data that form low-dimensional, arbitrary shapes or paths through a high-dimensional space. Instead of developing new algorithms appropriate for these problems, some authors proposed to obtain better solutions going back to stage one of clustering, i.e., by finding better ways to measure similarities between datapoints. In some sense, this view shares the spirit of kernel methods [7], looking for solutions to new problems by using appropriate new metrics together with well-known pattern analysis algorithms.

Excluding the classical metrics, like Euclidean or Pearson correlation, one of the most used metrics for clustering is the Gaussian or RBF metric [7], which we discuss in detail later in this paper. Fisher and Buhmann, introduced in two papers [8,9] another of such solutions, a metric we call the Path Based Metric (PBM). This metric is a derivation of a clustering method (Path based clustering) based on graph theory, which starts by assuming that, if two points belong to the same cluster, there should be a path between them, going over other points in the same cluster, such that all distances in that path are small. They consider then that the length of a given path can be defined as the maximum distance between two points in the path. Following, they define the distance between two points as the minimum length among all the paths that connect the two points.

The idea of measuring distances based on "neighbour of a neighbour" relations is the same that Bayá and Granitto [10] used in their Penalized k-nn graph (PKNNG) based metric. The main differences are that PKNNG uses a k–nearest–neighbour graph, instead of a fully–connected one and gives a less extreme value to the maximum edge. In fact, PKNNG considers all the edges in the path when it measures distances, but gives an exponential penalization to edges that lie in low-density regions. Because of these improvements, PKNNG has a great potential to deal with noisy, high dimensional situations [11]. Unfortunately, PKNNG in its current form has also two drawbacks. First, there is no clear procedure in the definition of PKNNG to deal efficiently with outliers in the problem. According to the actual procedure, graph edges connecting with outliers receive an exponential penalization to their weights. Subsequently and with high probability any clustering method will assign the outlier data-point to an isolated cluster. The other potential problem with PKNNG is that the number of neighbours K considered in creating the k-nn graph is not determined by an automatic procedure. Even tough the authors of the method discussed the problem and suggested that any low K value would show good performance, we found that in some cases a particular K value could lead to a better clustering solution.

In Section 2 of this paper we discuss a new, direct and simple strategy to handle outliers in the PKNNG metric. Also, we introduce an accurate method to obtain the optimal K value from a given range, using the Silhouette index originally proposed by Rousseeuw [12]. Then in Section 3 we compare the improved version of PKNNG with the original one [10], showing the good performance of the proposed solution. Finally, in Section 4 we compare several graph-based

metrics in a set of high dimensional problems coming from very different domains, including DNA microarrays and face and digits image recognition problems.

2 The PKNNG Metric

The PKNNG metric [10] is a two-step procedure. First, the method searches the original dataset space for locally dense (connected) structures using the knn-graph, with a low K. In the ideal case this process will end with a connected subgraph corresponding to each cluster but in the real case, when working with finite noisy samples, there are usually too many separated structures, typically more than one for each real cluster. In the second step penalized edges are added to the graph in order to fully connect it, and then an appropriate algorithm is used to measure distances in the connected graph.

First step: knn-graphs. As explained by Tenenbaum, de Silva and Langford [14], in a curved manifold the geodesic distance between neighbouring points can be approximated by the Euclidean input space distance. For distant points, in contrast, geodesic distances are better approximated as a path of short segments connecting neighbouring points. To this end, a knn-graph of the data is constructed, i.e. the graph with one vertex per observed example, arcs between each vertex and its k nearest neighbours, and with weights equal to the Euclidean distance between them. As the algorithm looks for dense subgraphs, at the end of the process all outliers are eliminated from the graph. An arc is considered an outlier if it is not reciprocal (i.e. one of the vertex is not a k-nn of the other) and the length of the arc is an outlier of its distribution (i.e. if it is larger than the 3rd quartile plus 1.5 times the inter-quartile distance).

Second step: connecting subgraphs. As we discussed before, the first step will usually generate separated structures, typically more than one for each real cluster. Having in mind that those subgraphs not always correspond to different clusters, it is important to be able to evaluate distances among points lying in different subgraphs. With that purpose, a number of edges are added to the graph in order to fully connect it. The key point of the PKNNG metric is the penalization of the weights of these new connecting edges [11]. An exponential factor of the form $w = d\ exp(d/\mu)$ is used, where w is the graph weight corresponding to the added edge, d is the Euclidean distance between the points being connected by that edge and μ is set as the mean value of all the k-neighbours distances considered in the first step. As explained by the authors [11], subgraphs corresponding to the same cluster can be connected with a relatively small cost when using this weighing scheme, because connections in the same order of magnitude of μ will get a low penalization. On the other hand, edges connecting distant subgraphs will be strongly penalized. Following the author's discussion, we use here the MIN-SPAN strategy to connect the subgraphs. In this scheme a minimum number of edges is added to the graph, each of them of minimum length to obtain a fully connected graph. For a detailed discussion of other penalizations and other connection strategies we refer the reader to Baya and Granitto [11].

2.1 Managing Outliers

Outlier datapoints are usually considered as problematic for clustering methods. The most usual strategy to deal with them is to identify and separate all the outliers first, then cluster the remaining, compact points and, as a final step, merge them to the most similar cluster. Other, less useful strategy is to give no special treatment to the outliers, letting them form individual clusters, which have the inconvenient of artificially rising the number of clusters found in the solution. The basic PKNNG metric eliminates edges connecting to outlier from the original k-nn subgraphs, thus in practice all the outliers are merged to the final graph via highly penalized edges. As a consequence, any clustering method working on PKNNG based distances has a strong tendency to separate all the outliers and use them to form individual clusters, which is an unwanted result.

This problem can be easily solved introducing a simple modification to the PKNNG metric. The idea is to prevent the method to penalize the edges connecting isolated points (outliers) but to keep it penalizing edges between disconnected structures. In order to do this, we include a simple constraint to the connection (second) step in PKNNG: any added edge connecting to a group formed by less than T points will not be penalized. The threshold T is a low value, typically 1 or 2, related to what can be considered as an outlier in the problem under analysis. A higher value can be used in some particular cases, for example when there is previous knowledge suggesting that very small clusters have to be avoided. Eliminating the penalization, most outliers will be merged to the most similar cluster, which is a more desirable solution to this problem.

2.2 Setting of K

The authors of PKNNG [11] proposed to use a low K value, from 3 to 7. In general, very low K values will produce one-dimensional, very unconnected structures, with no relation to the real data structure. On the other hand, big K values can add "shortcuts" to the graph, preventing it from producing geodesical distances between distant points. Though the range of values for K is clearly limited, we found in previous experiments that there is still some dependence on the PKNNG results with the setting of K, making desirable to define an appropriate way to choose an optimal value for K. To this end, we propose here to use the Silhouette index, introduced by Rousseeuw [12]:

$$s(i) = \frac{b(i) - a(i)}{max(b(i), a(i))}, \tag{1}$$

where $a(i)$ is the average distance from object i to all other objects in the same cluster and $b(i)$ is the minimum of the average distance of object i to all objects in each one of the other clusters. From the definition it is easy to see that $-1 < s(i) < 1$. A $s(i)$ value close to 1 means that point i was correctly clustered, while a $s(i)$ value near (–1) means that point i should belong to other cluster. The average silhouette (AS from here on) has been previously used as an accurate indicator of the goodness of a clustering solution [15,13], in particular in the

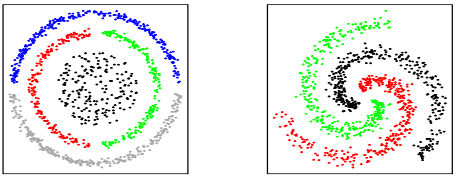
evaluation of solutions with different number of clusters. We propose here to also use AS as a measure of the quality of the metric produced by PKNNG when using different K values. As we show in the examples in the next section, there is a clear correlation between the AS and the goodness of the clustering solutions (according to the cRand index) also in this case.

3 Evaluation of the Improved PKNNG Metric

The idea behind using improved metrics in clustering is that the complexity of the problem should be filtered or solved by the metric, and thus any simple clustering algorithm would be enough to solve the resultant simplified problem. In all the experiments in this work we use the PAM clustering algorithm [15]. The faster K-means algorithm cannot be used in this case, because we are not working directly in a vector space and hierarchical methods are much more time-consuming than PAM. The problem of finding the right number of clusters for a given problem is out of the scope of this work. According to this, in all cases we assume that we are given in advance the number of clusters to find. In the artificial datasets case we clearly know the right cluster for each data-point (the golden rule of the problem) and the number of clusters involved. In the case of a real-world dataset, we consider that each cluster corresponds to one of the classes in the problem, i.e., that the class of each data-point is the golden rule for clustering, and that the number of classes in the problem is the number of clusters to find. In all the reported experiments we use the Euclidean metric as baseline for all other metrics. Also, we always use the corrected Rand Index (cRand) to compare the solution of the clustering method to the golden rule.

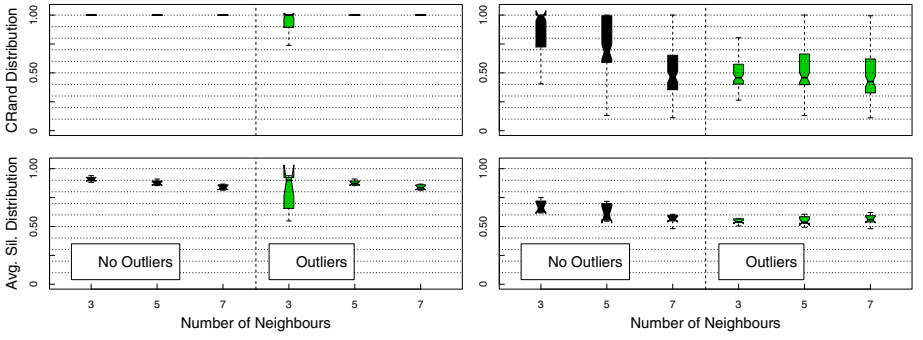
3.1 Empirical Evaluation

In Figure 1 we present the two artificial datasets used in this work. The three rings dataset, Panel a, is a 1200 datapoints, five clusters problem with difficulties associated with the curved shape and the lower density of the central cluster. The three spirals dataset, Panel b, is a 1050 datapoints, three clusters problem



(a) Three Rings (b) Three Spirals

Fig. 1. The two artificial datasets used in this work: a) The three rings dataset, with 5 clusters. b) The Three spirals dataset is a 3 arms spiral where each arm is a cluster. In both panels a different color is used for each cluster.



(a) Three Rings (b) Three Spirals

Fig. 2. Results on the two artificial datasets, Three rings on the left column and Three spirals on the right column. In both cases, top panels show the cRand distribution measured over 100 realizations and the bottom panels show the AS for the same clustering solutions. In all panels we evaluate $K = \{3, 5, 7\}$ and two outliers treatments: $T = 1$ (on the left) and $T = 0$ (on the right).

with curved shapes and changing densities in all clusters. We generated 100 realizations of each dataset, applied the PKNNG metric (with two values for the outliers threshold, $T = 0$ and $T = 1$) and clustered each realization. Finally, we computed AS and cRand for each solution.

In Figure 2 we show a comparison of the results obtained with the two different outliers treatment. Both top panels show the cRand distributions obtained for three different K values, for the two artificial datasets. It is clear from figure 2 that the new method proposed in this work for outliers handling ($T = 1$) shows a better performance than the previous method ($T = 0$), in particular for the Three Spirals dataset. It is interesting to note that the bigger improvements correspond to the $K = 3$ case on both artificial datasets. This minimum K is the most able to reproduce complicated structures, but its low connectivity has the drawback of easily producing outliers in the graph. Our new strategy for outliers handling allows PKNNG to increase its performance by using lower K values.

In the bottom panels we show the corresponding AS distributions for the same problems. In the $T = 1$ case the correlation between the cRand and AS figures for different K values is remarkable. On the other hand, in the $T = 0$ case the correlation is good but lower than in the previous situation. Overall, the analysis of the experiments on artificial datasets supports our proposal that the AS is an accurate indicator of the performance of the PKNNG metric under different K values.

We also applied the same analysis to the 6 real-world datasets considered in the next Section, with similar results. For lack of space we only show here the results corresponding to the UMIST faces recognition dataset [17]. This problem has 575 examples of facial images from 20 different people (112 by 92 pixels,8-bit gray-scale which we scaled to zero mean and unit standard deviation). We

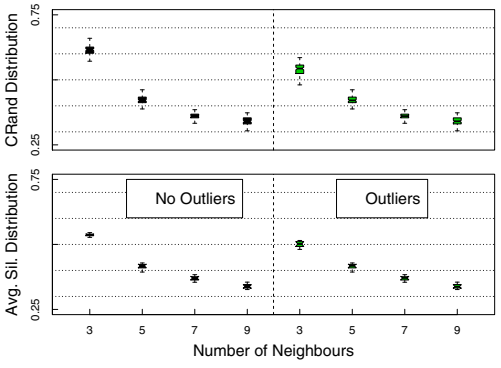


Fig. 3. Results on the UMIST faces dataset. The arrangement is similar to Figure 2.

generated 100 subsamples of the dataset, removing at random 2.5% of the datapoints each time, to generate diversity but maintaining enough samples from each class. In Figure 3 we show the corresponding results. The picture is qualitatively similar to the results on artificial data. There is a clear improvement in cRand values when using the new handling of outliers ($T = 1$). Also, there is a high correlation between AS and cRand distributions, and lower K values show a better performance in this problem.

4 Comparison of Improved Metrics on Real-World Datasets

We compare both versions of the PKNNG metric with two other interesting metrics. First, we consider the PBM, already discussed in the Introduction, which has some similarities with PKNNG and showed to perform particulary well in problems involving images [8,9]. PBM has no free parameters to set. Second, we also consider the well known RBF metric. This metric is based in the RBF inner product $\langle x, z\rangle = exp((\|x - z\|)^2/\sigma^2)$, using the property of a kernel vector space that relates kernel to distance, $\|x - z\|^2 = \langle x, x\rangle + \langle z, z\rangle - 2\langle x, z\rangle$. This metric has a free parameter, σ, that acts as a global scale for the solution. Following Caputo et al.[16], in all cases we set σ to the median of the Euclidean distance among all pairs of points in the dataset. For the PKNNG metric we select the K value for the number of neighbours analyzing the AS distributions, as discussed in the previous sections (data not shown for lack of space). We also include in this comparison the use of the plain Euclidean metric, in order to establish a baseline to show the relative benefits provided by the use of improved metrics in clustering.

We consider 6 real world datasets with very different characteristics. Yeast [18] is a biological dataset including 209 genes studied across 79 conditions with five functional groups where only four of them can be distinguished from their genomic profile (209 samples, 79 features, 4 classes). Central Nervous System tumors (CNS) [19] is a typical gene expression dataset, with 1000 genes (features)

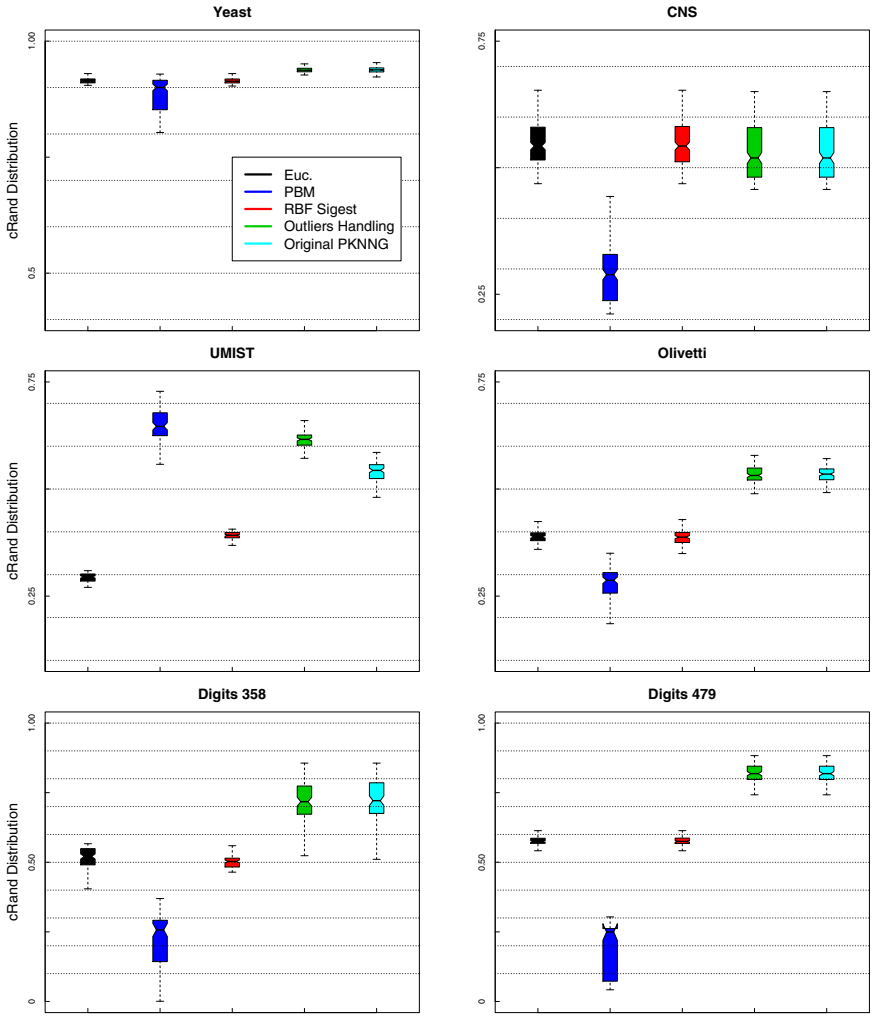


Fig. 4. Comparison of 5 metrics on 6 Real-world datasets. Results show distributions of cRand values for each problem.

evaluated in 42 cases including 5 types of tumorous tissue (classes). We also included two face recognition clustering problems, the UMIST dataset described in the previous Section and the Olivetti faces dataset[1]. The latter has 400 images of 64 by 64 pixels (4096 features) in 8-bit gray-scale (we applied the same normalization as in the UMIST case) of 40 different people (classes). Finally, we use two subsets of the MNIST[2] digits collected by LeCun and Cortez [20]. Digits are presented as 8-bit gray-scale images of 20x20 pixels (400 features).

[1] Available at http://cs.nyu.edu/~roweis/data.html
[2] Available at http://yann.lecun.com/exdb/mnist/

From the whole 60000 samples dataset and ten classes we took two subsets with 3 classes each, the first one including 18065 instances of digits $\{3, 5, 8\}$ and the second one with 17900 images of digits $\{4, 7, 9\}$. Overall, CNS is a typical wide biological dataset (many more variables than examples) while Yeast is a more balanced but still complicated biological problem. Both face recognition datasets are high dimensional image clustering problems (10394 and 4096 features) with a low number of samples per class. Finally, we analyze two digits recognition problems, which are also high dimensional but has many more samples than dimensions (high-dimensional "tall" datasets). To avoid results by chance we generated 100 different stratified subsamples of each dataset. In both biological datasets we remove 5% of the datapoints each time, 2.5% in both faces problems and 25% in the two MNIST datasets.

Figure 4 shows the results for the 6 problems. If we compare both versions of PKNNG, $T = 0$ and $T = 1$, the results are almost similar in five out of the six cases, except for the UMIST dataset where there is a clear difference (in fact, in four out of six cases the medians of the distributions are higher for $T = 1$, but there are no noticeable differences). Comparing with other metrics, PKNNG ($T = 1$) obtains the best results in all but two datasets (CNS and UMIST), in which the PKNNG metric is equivalent to the best metric (RBF and PBM, respectively). All other metrics show more problem-dependent results, in particular PBM which shows the worst performance in four problems. It is interesting to note that the base Euclidean metric works well only in the two biological datasets.

5 Conclusions

In this work we considered improvements to the existing PKNNG metric for clustering, discussing a new strategy to handle outliers and an efficient method to select the optimal K for the k-nn graph, based on the minimum of the Average Silhouette. We tested our proposed methods on both artificial and real data, verifying the good performance of the modified metric. We found that in same cases, after eliminating the penalization on the outliers, PKNNG can use a lower K value leading to a better general performance of the method.

In a second stage we further compared the improved PKNNG metric with other proposals from the recent literature. Our experiments with six real-world datasets showed that the improved version of the PKNNG metric produces very good clustering results in all cases under analysis, being also more stable than other metrics.

We showed that the coupling of a simple clustering method (PAM) with a better metric (PKNNG) can easily outperform the typical results obtained with that method and the base Euclidean metric. This is an interesting research line in clustering, as the practitioners can keep the easy-of-use and interpretation of the simpler methods but obtaining improved performances. We are currently working in other ideas in this direction.

References

1. Franti, P., Virmajoki, O., Hautamaki, V.: Fast Agglomerative Clustering Using a k-Nearest Neighbor Graph. IEEE Trans. Pattern Analysis Machine Intelligence 28, 1875–1881 (2006)
2. Yang, M., Wu, K.: A Similarity-Based Robust Clustering Method. IEEE Trans. Pattern Analysis Machine Intelligence 26, 434–448 (2004)
3. Yu, J.: General C-Means Clustering Model. IEEE Trans. Pattern Analysis Machine Intelligence 27, 1197–1211 (2005)
4. Xu, R., Wunsch II, D.: Survey of Clustering Algorithms. IEEE Trans. on Neural Networks 16, 645–678 (2005)
5. Ben-Hur, A., Elisseeff, A., Guyon, I.: A Stability Based Method for Discovering Structure in Clustered Data. In: Proc. Pacific Symposium on Biocomputing, vol. 7, pp. 6–17 (2002)
6. Tibshirani, R., Walther, G., Hastie, T.: Estimating the Number of Clusters in a Dataset Via the Gap Statistic. J. of the Royal Statistical Soc. B 63, 411–423 (2001)
7. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
8. Fischer, B., Buhmann, J.M.: Path-based clustering for grouping of smooth curves and texture segmentation. IEEE Trans. Pattern Analysis Machine Intelligence 25, 513–518 (2003)
9. Fischer, B., Buhmann, J.M.: Bagging for Path-based clustering. IEEE Trans. Pattern Analysis Machine Intelligence 25, 1411–1415 (2003)
10. Baya, A., Granitto, P.M.: ISOMAP based metrics for Clustering. Inteligencia Artificial 37, 15–23 (2007)
11. Baya, A., Granitto, P.M.: Penalized K-Nearest-Neighbor-Graph Based Metrics for Clustering, Technical Report, Cifasis 0001.09, http://arxiv.org/abs/1006.2734
12. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65 (1987)
13. van der Laan, M.J., Pollard, K.S.: A new algorithm for hybrid hierarchical clustering with visualization and the bootstrap. Journal of Statistical Planning and Inference 117, 275–303 (2003)
14. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
15. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, New York (1990)
16. Caputo, B., Sim, K., Furesjo, F., Smola, A.: Appearance-Based Object Recognition Using SVMs: Which Kernel Should I Use? In: Proceedings of Neural Information Processing Systems Workshop on Statistical Methods for Computational Experiments In Visual Processing and Computer Vision (2002)
17. Graham, D.B., Allinson, N.M.: Characterizing Virtual Eigensignatures for General Purpose Face Recognition. Face Recognition: From Theory to Applications 163, 446–456 (1998)
18. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences USA 95, 14863–14868 (1998)
19. Pomeroy, S., Tamayo, P., et al.: Gene Expression-Based Classification and Outcome Prediction of Central Nervous System Embryonal Tumors. Nature 415, 436–442 (2002)
20. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE 86, 2278–2324 (1998)

Empirical Evaluation of Ranking Prediction Methods for Gene Expression Data Classification

Bruno Feres de Souza[1], André C.P.L.F. de Carvalho[1], and Carlos Soares[2]

[1] ICMC/Universidade de São Paulo, São Carlos, Brazil
{bferes,andre}@icmc.usp.br
[2] FEP/Universidade do Porto, Porto, Portugal
csoares@fep.up.pt

Abstract. Recently, meta-learning techniques have been employed to the problem of algorithm recommendation for gene expression data classification. Due to their flexibility, the advice provided to the user was in the form of rankings, which are able to express a preference order of Machine Learning algorithms accordingly to their expected relative performance. Thus, choosing how to learn accurate rankings arises as a key research issue. In this work, the authors empirically evaluated 2 general approaches for ranking prediction and extended them. The results obtained for 49 publicly available microarray datasets indicate that the extensions introduced were very beneficial to the quality of the predicted rankings.

1 Introduction

Traditionally, cancer diagnostics has relied on the tumor's morphological appearance and on the site of its origin. With the introduction of large scale gene expression profiling tools, such as microarrays [1], researchers have earned the opportunity to better understand the tissues being treated and to achieve a finer molecular distinction between health states. In order to make the proper analyzes and acquire qualitatively interesting information from the biological data generated in this type of experiments, one usually employs computational tools. Specifically, Machine Learning (ML) based approaches have been largely considered, mainly due to their ability to automatically extract patterns from data.

The choice for the ML approach to use on a given classification scenario is not a trivial task. The current rules of thumb for algorithm selection rely on costly trial-and-error procedures or clinicians, who need to analyze microarray data more direct and cost-effectively. Thus, in practice, the algorithms to be considered are basically determined by the familiarity of the user with the method rather than the particularities of the data and the algorithms themselves. This may lead to sub-optimum results, compromising the whole experimental setup.

In an attempt to overcome these problems, meta-learning has been frequently used for algorithm recommendation [2]. In many approaches, the suggestions provided to the user consist of a single algorithm. However, this may not be the most adequate form of recommendation for this task, since it does not provide

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 194–203, 2010.

any further guidance when the user is not satisfied with the results obtained with the recommended algorithm. Despite executing all the algorithms may lead to a high computational cost, frequently, the available computational resources are sufficient to run more than one of the available algorithms. If recommendation indicates the order in which the algorithms should be executed, the user can execute as many algorithms as possible, thus increasing the probability that a satisfactory result will be obtained. Therefore, the problem of algorithm recommendation should be tackled as a ranking prediction task [2].

In this paper, the authors revisite the work of [3] regarding the use of meta-learning in the context of gene expression data classification and propose an extension to the kNN ranking method employed there. The so called wkNN is based on the intuition that examples in the training set that are particulary closer to the test examples should have a higher influence in the construction of recommendation. The experiments show very positive results for the wkNN when 2 kernels are considered.

Within the same classification context, the authors also investigated if ranking trees [4] would be able to accurately predict rankings in a restricted domain. The performance behavior of such methods has not been fully evaluated. While [4] reported the ability of ranking trees to outperform kNN ranking method and default ranking in a number of situations, [5] presented a comprehensive set of experiments that contradict those claims. They argued that different experimental setup could have led to divergent results. Thus, in this paper, some experiments were conducted to verify if ranking trees could be successfully employed with gene expression data. Furthermore, the authors introduced the concept of bagging of ranking trees with promising experimental results.

This document is organized is follows. In Section 2, the general architecture of the meta-learning system employed is explained. Section 3 discusses the ranking prediction methods developed. Section 4 presents the experimental results obtained. Finally, Section 5 draws the conclusions of this work.

2 Meta-learning

For the purposes of this work, meta-learning consists of using a learning algorithm to model the relation between the characteristics of learning problems and the relative performance of a set of algorithms [2]. It usually envolves three steps: (1) the generation of meta-data; (2) the induction of a meta-learning model by applying a learning algorithm to the meta-data; and (3) the application of the meta-model to support the selection of the algorithms to be used in new datasets. Next, the authors summarize these steps. For a more thorough description the reader is referred to the work of [2] and references therein.

2.1 Meta-data

In order to perform meta-learning, it is necessary to have adequate meta-data, which are analyzed by a learning algorithm, resulting in meta-knowledge. Meta-data describe various characteristics present in a group of problems, typically

provided in the form of datasets, and the performance of a set of algorithms when they were applied to these datasets. In practice, meta-data consist of a set of meta-examples, each one of them corresponding to one dataset. The meta-examples are represented as a number of input attributes, encoding the characteristics of the datasets, and the corresponding target attribute, encoding the performance of a set of algorithms. The input attributes, also known as meta-features, are measures that characterize the datasets. These measures represent generic properties of the data that are expected to affect the performance of the ML algorithms. The most common set of meta-features is the so called GSI (General-Statistical-Information), proposed in the Statlog Project [2]. Here, the authors considered a subset of 10 of these measures which have been successfully employed in the gene expression context (see [3] and references therein). The target attributes represent the relative performance of the set of ML algorithms on the dataset. This is a flexible form of recommendation which allows the user to investigate different sets of ML algorithms.

2.2 Learning Rankings

In general, a ranking represents a preference function over a set of items [2]. Here, the items are the ML algorithms considered for gene expression data classification, while the preference function expresses the expected performance of the algorithm in a dataset. Thus, if a given algorithm is supposed to perform better than other on a particular dataset, it should be assigned a higher rank. In the context of meta-learning, the target of the meta-examples consists of a ranking of the ML algorithms. The task at hand is to learn the underlying relation between the meta-features and the target rankings in the meta-data. This learning problem is similar to the problem of supervised learning. However, while supervised learning is concerned with learning how to associate a single target value (which is an ML algorithm in meta-learning) with each example, meta-learning aims to predict the ranking of all the possible target values for each example.

Two popular approaches presenting distinct points of view for ranking construction were employed in this work and will be explained in Section 3. Alternatives can be found elsewhere [2].

2.3 Evaluation and Application

After a meta-model is created using a given ranking method applied to set of meta-data, it is necessary to produce evidence to the user that meta-learning is able to generate accurate predictions. An approach for this is to use Leave-one-out Cross Validation (LOOCV), which iteratively, for each meta-example, computes the accuracy of the predicted ranking using a meta-model obtained on all the remaining meta-examples [2]. To measure ranking accuracy, the authors have used Spearman's Rank Correlation Coefficient RS, which simply evaluates the monotonicity of two sets of values, i.e., if their variations are related. If they tend to increase or decrease together, the variables are positively correlated.

However, if one tends to increase while the other decreases, then they are negatively correlated. This coefficient ranges in the interval $[-1, 1]$. Another way to measure ranking accuracy exploits the fact that, in the algorithm recommendation setting, the user tends to prefer the algorithms better placed in the ranking. Thus, the Log Ranking Accuracy (LRA) [2] assigns greater importance to higher ranks. Unless stated otherwise, RS is the primary metric of comparing rankings in this work.

To determine whether the accuracy of some particular recommended ranking can be regarded as high or not, a baseline method is required. In ML, simple prediction strategies are usually employed to set a baseline for more complex methods. For instance, a baseline commonly used in classification is the most frequent class in the dataset, referred to as the default class. The baseline is typically obtained by summarizing the values of the target variable for all the examples in the dataset. In ranking, a similar approach consists of averaging all the target rankings in the training set of the meta-data. The ranking obtained is named the default ranking (DR).

3 Ranking Methods

3.1 Nearest Neighbor Based Ranking

An algorithm that has previously been adapted for learning rankings and applied to the meta-learning problem with successful results is the k-Nearest Neighbors (kNN) algorithm. It consists of selecting the k closest examples in the training set (the Euclidean distance was considered here) and then aggregating the corresponding target rankings to generate the prediction for the new example. A simple approach is to aggregate the k target rankings with the Average Ranks (AR) method [2], which equally considers all neighbors during the averaging process.

In the present work, the authors propose a modification of the traditional kNN to allow an uneven contribution of the neighbors, according to their distance from the test example. This extension is a straightforward adaptation of the *weighted k-Nearest Neighbor* [6], designed for classification with single target ,to deal with target attributes in the form of rankings. It will be named here wkNN ranking method.

For constructing the ranking for a test example, the wkNN method initially selects the k closest examples with distances d_i, $i = 1, ..., k$ and standardize d with respect to the $k + 1^{th}$ neighbor, such that $\forall i, d_i \in [0, 1]$. Through the use of a kernel function $K(\cdot)$, these distances are converted into similarity measures, which will latter be employed as weights for the respective examples. Typical kernel functions include [6]:

- Retangular: $\frac{1}{2} \cdot I(|d| \leq 1)$
- Triangular: $(1 - |d|) \cdot I(|d| \leq 1)$
- Epanechnikov: $\frac{3}{4}(1 - d^2) \cdot I(|d| \leq 1)$
- Gauss: $\frac{1}{\sqrt{2\pi}} e^{(-\frac{d^2}{2})} \cdot I(|d| \leq 1)$

At this point, the k target rankings can be aggregated to generate a prediction. For such, let $R_{i,j}$ be the rank of base-algorithm a_j, $j = 1, ..., n$ on dataset i, where n is the number of algorithms. The weighted average $w\bar{R}_j$ for each a_j is given by:

$$w\bar{R}_j = \frac{\sum_{i=1}^{k} w_i * R_{i,j}}{\sum_{i=1}^{k} w_i} \quad (1)$$

The final ranking is obtained by ordering the weighted average ranks and assigning ranks to the algorithms accordingly. When the Rectangular kernel is used, the wkNN performs with no example weighting and thus behaves exactly like the ordinary kNN ranking method [2].

3.2 Tree Based Ranking

While the Nearest Neighbors based approach relies on considering the ML algorithms individually and later combine their average relative positions to construct rankings, tree based methods work with all algorithms at once. Thus, they can exploit relations between the ML algorithms that other approaches cannot. This could, potentially, lead to better predictions. Another benefit is that it can clearly express information about the problem, because their structure is relatively easy to interpret even for non expert users, as biologists or physicians.

In the context of algorithm recommendation, trees were introduced by [4]. They employed a variant called Predictive Clustering Tree (PCT), which is fully implemented in the Clus Software [4]. It works by identifying partitions of data such that a hierarchy of clusters is generated. Individuals placed in the same cluster are expected to be similar according to some criteria. The Top-Down Induction of Decision Trees (TDIDT) algorithm can be used to iteratively splits data into smaller subsets that are increasingly more homogeneous in terms of the target attributes. In the case of predicting rankings, the target attributes are the relative order of the ML algorithm for each example. Thus, PCT tries to minimize the intra-cluster variance of the partition with respect to the rankings of the examples. This goal can be seen as an approximation to maximizing Spearman's correlation of the rankings of the examples inside the same cluster.

Another hypotheses the authors try to validade here is if bagging could be used together with PCT, given it has been shown that this ensemble technique can give substantial gains in predictive performance, when applied to an unstable learners such trees [7]. At a glance, bagging constructs different classifiers by making bootstrap replicates of the training set and using each of these replicates to induce one classifier. Each bootstrap sample is obtained by randomly sampling training instances, with replacement, from the original training set, until a number of instances equal to the number in the original training set is obtained.

4 Experimental Results

The main goal of the experiments reported here is to assess the performance of the two general approaches for ranking construction presented earlier. For

such, the evaluation and application guidelines presented in Section 2 were employed. The problems used to generate the meta-data come from 49 publicly available microarray datasets. They were classified using 7 ML algorithms [1]: DLDA, DQDA, PAM, 3NN, SVM-L, SVM-R and RF, which were evaluated according to the bootstrap .632+ technique. Due to space constrains, the interested read is refereed to the work of [3] for a full description of the datasets and the algorithms considered.

4.1 Nearest Neighbors Based Ranking

Figure 1 shows the results of experiments using the wkNN ranking method with the 4 kernels discussed in Section 3 and the default ranking. The mean predictive accuracies were calculated for the 49 datasets, with the number of neighbors varying over $k = 1, ..., 30$.

Initially, the performances for all kernels increase as k increases, since predictions with very few neighbors may be less robust to noise data and a larger neighborhood can mitigate this effect. However, as the size of the neighborhood increases, the prediction is affected by neighbors that are increasingly less similar to the current test example and, thus, performance of the wkNN technique decreases. This degradation in performance occurs first with the rectangular kernel. After reaching its maximum value with $k = 4$ (0.698 mean accuracy), it presents an erratic descendent tendency towards the default ranking (0.608 mean accuracy), indicating that it is very sensitive to the values of k. This behavior is, to some extent, shared by the Gauss kernel, although it presents sightly better results over the entire range of k. Triangular and Epanechnikov kernels, on the other hand, achieved more clear improvements over wkNN Retangular. From $k > 6$, they are always superior, with advantage to the triangular kernel. Their best performances are, respectively: 0.723 ($k = 13$) and 0.716 ($k = 12$). For $13 < k \leq 30$, they present a smooth decay in mean accuracy. The superiority of the triangular kernel, specially for large values of k, may be better understood by inspecting its shape according to the definition presented in Section 3. One can realize that the weight of neighbors in the construction of the ranking rapidly decreases with their distance. Thus, distant neighbors soon become less important. In the case of Epanechnikov kernel, the detriment in the influence of far neighbors is more even. For the Gauss kernel, it is flatter.

To deal with the need to select an appropriate value of k, the authors employed an inner LOOCV procedure to select the best numbers of neighbors k^* for each dataset, over $k = 1, ..., 15$. The k^* with the largest mean accuracy was chosen. For the Rectangular, Triangular, Epanechnikov and Gauss kernels, this approach selected in average, respectively: $k = 5.61$ (1.85 in standard deviation), $k = 12.51$ (1.37), 10.63 (1.94) and $k = 8.08$ (1.75) neighbors, yielding mean accuracies of 0.671, 0.722, 0.710 and 0.689. Interestingly, the numbers k^* are in the optimal

[1] In [3], the authors employed a 8th classifier, penalized discriminant analysis (PDA. After a careful analysis of the results presented on that paper, it was noted that PDA and SVM-L performances were essentially the same. Therefore, PDA was not considered in the presente work.)

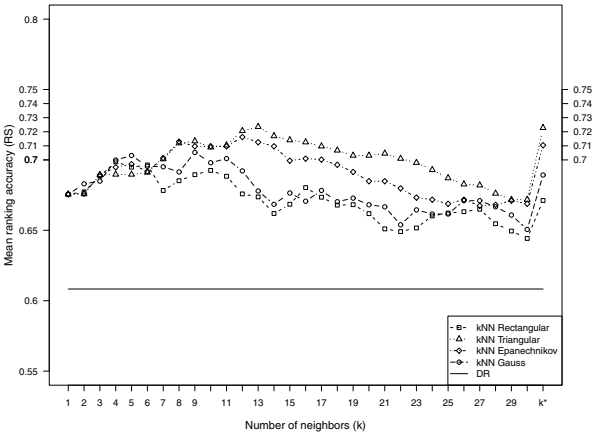


Fig. 1. Mean accuracies for wkNN with 4 kernels varying the number of neighbors

range for all kernels (see Figure 1). Nevertheless, the Rectangular kernel seems to overfit the data with k^*, while the Triangular kernel almost reaches its best value over $k = 1, ..., 30$. As can be noted, the selected numbers of neighbors are larger for the 3 more sophisticated kernels. This indicates that, indeed, they are more robust to the choice of k, due to their weighting capability.

When comparing wkNN (using k^*) with the default ranking, Triangular, Epanechnikov and Gauss kernels led to statistically superior accuracies, according to Friedman's test with 95% confidence and Dunn's multiple comparison procedure with 90% confidence. Considering the 4 kernels, there were no significant differences. Thus, for the overall comparison of the ranking methods in Subsection 4.3, only wkNN with Triangular kernel and k^* will be considered.

4.2 Tree Based Ranking

In this subsection, the authors compare the performance of PCTs and bagging of PCTs. As stated by [4], results using the Clus software to induce PCTs are very sensitive to the value of the `Ftest` parameter, which controls the rate of splits in a tree and ranges from 0 to 1. Values close to 0 leads to few splits and possibly premature convergence, while the value 1 generates trees with the maximal size. Following their recommendation, `Ftest`=1 was considered. As can be seen in Figure 2, PCT did not perform very well in constructing rankings for the problem investigated here, with a mean accuracy of 0.625, just above the performance of the default ranking, 0.608. In the context, such large PCTs seem to overfit the data. In attempt to overcome this, the authors applied the experimental modification of the M5 pruning method for multi-target regression trees as implemented in Clus (for parameter `M5Multi`=1). With pruning, the mean accuracy of PCT was increased to 0.658.

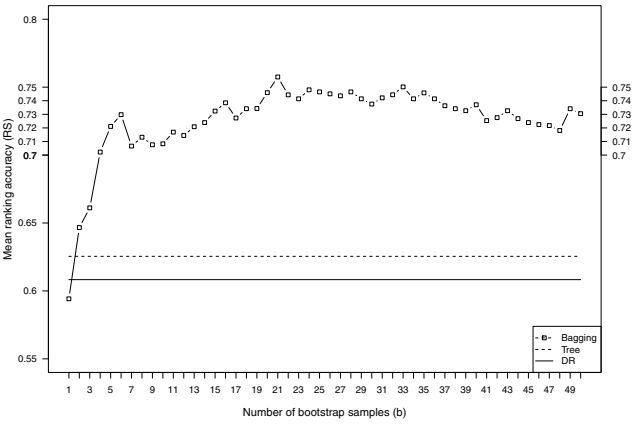


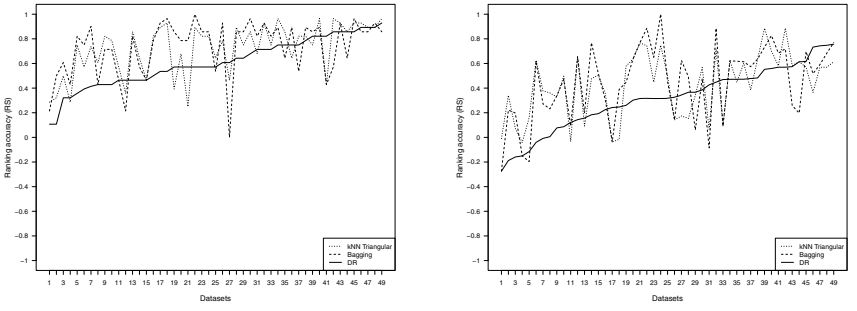
Fig. 2. Mean accuracies for bagging varying the number of bootstrap samples

The results with bagging presented in Figure 2 show more competitive improvements. In this figure, the number of bootstrap samples varies over $b = 1, ..., 50$. Initially, the rankings obtained are not very accurate. This occurs due to the reduced number of unique data examples in the training set, roughly 63.2% of the total data [7]. Thus, less information becomes available for the induction of the tree. At $b = 1$, the mean accuracy of bagging is 0.594, lower than the result for PCT. However, as soon as b increases, mean accuracies increase together, because predictions from different, non-correlated, trees are combined. While $b \leq 21$, bagging presents various local minima in performance, but with a clear ascendent tendency. The best mean accuracy of 0.757 occurs at $b = 21$. Afterwards, mean accuracies tend to become more homogeneous, since it is expected that with more bootstrap samples, bagging is less affected by the randomness of the sampling procedure and thus achieves more stable results. Based on this, the authors will consider only bagging with $b = 50$ (0.730 of mean accuracy) for the analysis presented in Subsection 4.3. Furthermore, when comparing bagging with $b = 50$, PCT and the default ranking, the former is statistically superior to its competitors, according to Friedman's and Dunn's tests. PCT and default ranking performed similarly.

4.3 Overall Comparison

Figure 3(a) compares the performance of wkNN with Triangular kernel, bagging and the default ranking. It shows the ranking accuracies (using Spearman correlation) of the 3 methods for the 49 datasets considered here. To improve readability, the datasets are sorted using the default ranking performance. As can be seen, wkNN and bagging were able to generate rankings that are more correlated to the ideal rankings more frequently than the default ranking. The mean (and standard deviations of) ranking accuracies of wkNN, bagging and the default ranking were, respectively: 0.722 (0.208), 0.730 (0.22) and 0.608 (0.200).

According to Friedman's and Dunn's tests, the 2 methods are statistically superior to the default ranking. Despite of this, they sometimes produce less accurate rankings. Drops in the performance of a metalearning method may be related to 1) unusual (relative) performance of the algorithms, 2) inadequacy of the meta-features used or 3) insufficient amount of meta-data in the region of the test dataset. The first reason is unlikely in this case, because it would probably also imply a drop in the performance of the default ranking (which is fixed). Therefore, it is necessary to investigate the remaining questions, which will be left for future work. According to the same tests, the differences between wkNN and bagging are not significant.



(a) RS correlation (b) LRA correlation

Fig. 3. Accuracies over all datasets considering wkNN Triangular and bagging

Figure 3(b) presents the same type of comparison using the LRA measure instead of the spearman correlation to assess the accuracy of the generated rankings. The outcome is similar. Mean (and standard deviation) for wkNN, bagging and default ranking are, respectively: 0.428 (0.259), 0.446 (0.306) and 0.315 (0.259). Although the values obtained are not directly comparable to the values obtained using Spearman's coefficient, they indicate that all the methods do not predict the top ranked algorithms very accurately. This is not a surprise, since the meta-learning methods used do not focus on the top ranks. This results confirm that there is a need to develop ranking methods that focus on the most important ranks. Applying statistical tests like before, wkNN and bagging were superior to the default ranking and equal to each other, indicating that the 2 methods are better at predicting the top ranks than the default ranking.

5 Conclusions

In this paper, an evaluation study was performed with two common ranking prediction methods: kNN and trees. The specific domain of gene expression data classification was considered. Ordinary kNN performed well, producing more accurate results than the default ranking method. However, two of its weighted

versions, wkNN Triangular and wkNN Epanechnikov yielded very good performance. On the other hand, the performance of tree ranking methods based on PCT was disappointing. Nevertheless, their ensemble counterpart fully compensate the expectation for the tree based methods, producing rankings more accurate than the best wkNN variant. As future works, the authors plan to investigate more kernels to use together with the wkNN and to examine the potential explanatory relevance of the model trees generated with the bagging approach.

Acknowledgements

This work was partially supported by FCT project Rank! (PTDC/EIA/81178/2006) and Brazilian funding agencies CAPES, CNPq and FAPESP.

References

1. Russell, S., Meadows, L., Russell, R.: Microarray Technology in Practice, 1st edn. Academic Press, San Diego (October 2008)
2. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. In: Cognitive Technologies. Springer, Heidelberg (2009)
3. Souza, B.F., Soares, C., de Carvalho, A.: Meta-learning approach to gene expression data classification. International Journal of Intelligent Computing and Cybernetics 2(2), 285–303 (2009)
4. Todorovski, L., Blockeel, H., Dzeroski, S.: Ranking with predictive clustering trees. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, pp. 444–455. Springer, Heidelberg (2002)
5. Rebelo, C., Soares, C., Costa, J.P.: Empirical evaluation of ranking trees on some metalearning problems. In: Chomicki, J., Conitzer, V., Junker, U., Perny, P. (eds.) Proceedings of the MPREF 2008, Held in Conjunction with the AAAI 2008, Chicago, Illinois (July 2008)
6. Hechenbichler, K., Schliep, K.: Weighted k-nearest-neighbor techniques and ordinal classification (2004)
7. Breiman, L.: Bagging predictors. Mach. Learn. 24(2), 123–140 (1996)

Defining a Problem-State Representation with Data Mining within a Hyper-heuristic Model Which Solves 2D Irregular Bin Packing Problems

Eunice López-Camacho[1], Hugo Terashima-Marín[1], and Peter Ross[2]

[1] Tecnológico de Monterrey,
Av. E. Garza Sada 2501,
Monterrey, NL, 64849 Mexico
{eunice.lopez,terashima}@itesm.mx
http://www.itesm.edu
[2] School of Computing,
Napier University
Edinburgh EH10 5DT UK
P.Ross@napier.ac.uk
http://www.napier.ac.uk

Abstract. In various approaches for combinatorial optimization, a problem instance is represented by a numerical vector that summarizes to some extent the actual solution state of such instance. Such representation intends to include the most relevant features related to the instance and the problem domain. The proper selection of these features has a direct impact on the performance of a hyper-heuristic. Previous approaches for hyper-heuristics have been relying on intuitive ways to determine the feature set, usually based on the domain knowledge of a particular problem. In this paper, a more general methodology for establishing an adequate problem-state representation is proposed. We chose the irregular case of the two-dimensional Bin Packing Problem (2D irregular BPP) and a GA-based hyper-heuristic model to test the methodology. As far as we know, this is the only hyper-heuristic model applied to the 2D irregular BPP and it has been successful when solving a wide range of instances. Our developed representation shows a significant improvement in performance with respect to a more conventional representation for the 2D irregular BPP.

Keywords: Evolutionary Computation, Hyper-heuristics, Optimization, 2D irregular Bin Packing Problem, Problem State Representation.

1 Introduction

A hyper-heuristic is used to define a high-level heuristic that controls single (low-level) heuristics. A hyper-heuristic should decide which single low-level heuristic to apply, depending on the given problem state which is summarized by a numerical vector called *representation* [1]. This idea of hyper-heuristic has been applied

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 204–213, 2010.

for solving the 1D and 2D Bin Packing Problem (BPP) [2], [3], [4], [5], [6], as well as the Constraint Satisfaction Problem (CSP) [7]. Other approaches have used representation of instances (and/or instance-states) based on selecting some relevant features; for example, the hyper-heuristic model that uses Case-based Reasoning for Course Timetabling Problems [8].

Much of the performance of a hyper-heuristic model may depend on the choice of representation of the problem state and the choice of the particular set of heuristics used [1]. We propose a new general methodology for producing a representation using data mining for feature selection and we tested it in the representation scheme within a hyper-heuristic approach for solving the 2D irregular BPP. The basic assumption is that similar problem instances would be associated with the same single heuristic. The objective is to characterize instances into *homogeneous* classes and relate them with the best single heuristic for each class. The idea behind hyper-heuristics is to associate each heuristic with the problem conditions under which it flourishes and hence apply different heuristics to different parts or phases of the solution process. That is why this work attempts to find the key features of a problem that makes instances suitable for one heuristic or another.

The 2D irregular BPP is considered as the experimental environment. For this problem, a hyper-heuristic solution model has been applied with encouraging results [6] in which some of the features selected for the state representation were taken from the 2D regular (rectangular) BPP [4] whose representation was influenced for the one-dimensional case as well [3]. Problems known to be hard have certain characteristics. In 1D-BPP, for example, the hard benchmark problems involve items whose weights are typically a significant fraction of the bin capacity, so that the difficulty arises when trying to find the subsets of items that are to reside in each bin [1]. This makes much sense in the 1D-BBP. In the 2D case, not only the size is important but also the shape; and in the 2D irregular BPP, some features linked with irregularity appear. Some of these features may be relevant in the representation of the problem state, some others may not. Which features are the most appealing for the representation of the problem state may not be obvious.

The article is organized as follows: Section 2 describes the 2D irregular BPP as well as the hyper-heuristic solution model used in the investigation. Section 3 presents the description of the proposed methodology and the development of a representation scheme for the 2D irregular BPP. Section 4 presents the experimental evaluation and the discussion of results. Section 5 points out the conclusion.

2 A Hyper-heuristic Solution Model for the 2D Irregular BPP

The Cutting and Packing Problem is among the earliest problems in the literature of operational research. Due to the extensive work done in this NP-problem, in 2006 Wäscher et al. [9] suggested a complete typology which is an extension of Dychoff's [10]. Given a set $L = (a_1, a_2, \ldots, a_n)$ of pieces to cut or pack and an

Table 1. Set of single heuristics

Selection	Placement
1. First Fit (FF)	1. Bottom Left
2. First Fit Decreasing (FFD)	Irregular (BLI)
3. First Fit Increasing (FFI)	2. Constructive
4. Filler + FFD	Approach (CA)
5. Next Fit (NF)	3. Constructive
6. Next Fit Decreasing (NFD)	Min Area (CAA)
7. Best Fit (BF)	4. Constructive Max
8. Best Fit Decreasing (BFD)	Adjacency (CAD)
9. Worst Fit (WF)	
10. Djang and Finch (DJD)	

infinite set of identical rectangular larger elements (called *objects*) with measures x_0 and y_0, the 2D-BPP consists of finding an arrangement of *pieces* inside the *objects* such that the number of objects required to contain all pieces is **minimum**. A feasible solution is an arrangement of pieces without overlaps and with no piece outside the object limits. A *problem instance* or *instance* $I = (L, x_0, y_0)$ consists of a list of elements L and object dimensions x_0 and y_0. An instance *state* is every intermediate phase in the solution process until all pieces are placed and a solution for the instance is found. The problem is called 2D irregular BPP when pieces are not rectangular [9].

2.1 Combining Heuristics with a Genetic Algorithm (GA)

Terashima et al. [6] present a GA-based method that produces general hyper-heuristics. As far as we know, this is the only hyper-heuristic model applied to the 2D irregular BPP, when tested with a large set of benchmark problems, produces outstanding results for most of the cases [6]; that is why, this is the solution model employed for the experimental environment.

Set of Heuristics - Heuristic approaches for the 2D BPP present at least two phases: first, the selection of the next piece to be placed and the corresponding object to place it; and second, the actual placement of the selected piece in a position inside the object according to a given criteria. In this investigation ten selection heuristics and four different placement heuristics were employed. For details of how these heuristics work see [6]. All possible combinations of selection and placement heuristics conform the *single heuristic repository* H, that is the set of possible actions available to solve an instance (see Table 1). Here, $|H| = 40$.

Representation of the instance state - Each instance to be solved by the hyper-heuristic is characterized by a numerical vector that summarizes some of its relevant features. Each numerical term of the vector quantifies an aspect of the instance at hand. According to this numerical vector, the hyper-heuristic decides which single heuristic to apply every time. The numerical representation is applied for *complete instances* to be solved (where no piece has been placed yet) as well as for *instances partially solved* (where some pieces have already been placed). That is, for a given problem instance, numerical representation is computed in every intermediate state until it is completely solved.

Table 2. Representation **1** of the instance state

Feature	Description
	Fraction of remaining pieces in the instance:
1	with high rectangularity, in the range $(0.9, 1]$.
2	with medium rectangularity, in the range $(0.5, 0.9]$.
3	with low rectangularity, in the range $(0, 0.5]$.
4	whose area is larger than 1/4 of the object area.
5	whose area is larger than 0.1 and up to 0.25 of the object area.
6	taller than the half of the object area.
7	taller than 0.25 and up to 0.5 of the object total height.
8	that remain to be packed.

In the work by Terashima et al. [6] eight features related to the problem domain were selected by intuition to represent an instance state (see Table 2). This is called Representation 1 in this research. The features were selected by analyzing various parameters present in irregular 2D pieces, such as rectangularity, area, length, hight and the percentage of remaining pieces to be packed. Rectangularity is a quantity that represents the proportion between the area of a piece and the area of a horizontal rectangle containing it.

Chromosomes - Each chromosome is composed of a series of *blocks*. Each block includes several numbers. All numbers in a block, except the last one, represent an instance state which is the numerical vector mentioned above. The label is the last number, which identifies a single heuristic (selected from the 40 possible combinations of selection and placement heuristics shown in Table 1). Given an instance and having computed its numerical representation, find the closest point in the chromosome (with Euclidean distance) and apply the single heuristic recorded on the label. This will produce a new state and the process is repeated until a complete solution has been constructed. The GA's task is to find a chromosome that is capable of obtaining good solutions for a wider variety of problems; the chromosome is the hyper-heuristic that we seek.

The Fitness Function - The quality of a solution, produced by either a single heuristic or a hyper-heuristic for a given instance, is based on the percentage of usage for each object:

$$Q = \frac{\sum_{i=1}^{No} U_i^2}{N_o} \quad . \tag{1}$$

where N_o is the total number of objects used and U_i is the fractional utilization for each object i. The result of the best single heuristic for each instance is stored (BSH). Now, each chromosome solves some instances and its fitness is computed as the average difference between the solution quality obtained by the chromosome with respect to the result given by the best single heuristic for every particular instance:

$$f = \frac{\sum_{k=1}^{m} (Q_k - BSH_k)}{m} \tag{2}$$

where BSH_k is the best quality solution obtained by a single heuristic for the k-th assigned instance, Q_k is the quality solution obtained by the hyper-heuristic for the k-th assigned instance and m is the number of instances solved so far. BSH_k and Q_k are computed using equation 1.

The available problem instances are divided into a training and a testing set. The GA is used with the training set only, until a termination criterion is met and a general hyper-heuristic has been evolved. All instances in both sets are then solved with this hyper-heuristic. For more details see [6].

3 The Proposed Representation Scheme

This section explains the proposed methodology for developing a representation scheme which uses data mining techniques to determine the feature set. The methodology is applied to the 2D irregular BPP.

3.1 Methodology for Developing a Representation Scheme

Data mining is the process of finding hidden patterns in large amounts of data. One of its main applications has been scientific discovery. The general idea in this analysis is to describe a methodology to find problem relevant features using a data mining approach.

Step 1. Solve each instance with each single heuristic from the heuristic repository H and compute a measure of performance.

Step 2. Register the performance of all heuristics for each instance as a vector in $\Re^{|H|}$, where $|H|$ is the size of the heuristics repository H. Normalize this vector, dividing by its length (to get length of one). Unit vectors are used to indicate direction, that is, they aim in the direction of the better heuristic for each instance.

Step 3. Classify all instances into homogeneous clusters according to their normalized performance measure for all heuristics. There are many clustering techniques, one of these is the k-means technique which is an old but popular clustering algorithm known for its observed speed and its simplicity [11].

Step 4. Determine a set of problem features or measures that may be relevant to the ability of the heuristics to solve each instance. For example, in BPP, average size of pieces may be related on how hard an instance is. Next steps will prune the list to keep just the more relevant features.

Step 5. Determine pairs or sets of equivalent features. It may happen that some couple or small sets of features are highly correlated (positive or negative) since they carry almost the same information. In this case, we can say we have equivalent features. The Pearson correlation coefficient is an immediate way to perform this. For every pair or set of equivalent features it is possible to choose just one of them which would *act* as the representative of the other(s) and *delete* the others reducing the total set of features.

Step 6. Among the set of the problem features, select a subset which better predicts the instances clusters. This selection could be done through Multinomial

Logistic Regression (MLR) [12] which is a statistical method extension of the (binary) logistic regression when the categorical dependent outcome has more than two levels and can be used to assess the influence of explanatory variables (problem features) on the nominal response variable (cluster). The MLR model has to be run with all the features as independent variables. Since instances in each cluster tend to have better performance in the same single heuristics, those features that are more related with the clusters will be the same features that influence instances to be better solved for one heuristic or another.

The features selected by the last step are the ones most suitable to be used in the problem representation scheme.

3.2 Representation Scheme 2

In order to test the data-mining based methodology, the 2D irregular BPP is chosen. The same 541 instances and 40 single heuristics are used from [6]. Each instance was solved by each single heuristic and the performance measure was computed with equation 1. Heuristics performance for each instance was seen as a vector in a 40-dimensional space. Every vector was normalized. We applied the k-means algorithm for clustering the 541 available instances into five groups according to their normalized heuristic performance.

Nineteen numerical features were computed for each instance. A correlation analysis was performed to detect pairs or sets of features highly related, and thus equivalent. The criteria chosen was an absolute value of correlation index at least of 0.9. With this criteria, six features were eliminated from the analysis, since they carried what we could call *redundant information*. The remaining 13 features were used to predict instance cluster through MLR. This model predicts correctly the cluster of 80.0% of the instances. This is, by far, greater than the corresponding by chance. So, the selected features are indeed related with normalized heuristic performance. Nine out of the 13 features used in the model were significantly related with cluster membership (p-value < 0.05). We selected seven from this set. This features can predict to some extent the way single heuristics behave at instances of the 2D irregular BPP.

All selected features are not in the same order of magnitude. While one feature could be measured in a percentage (from 0 to 1), another feature could be measured in thousands of units, for example. To cope with this, we performed a linear mapping to a fixed scale to all feature values used in the instance state representation making each possible feature value to fall inside the range from 0 to 1. Features that are represented as percentages are naturally in that range (for example, feature 4 in Table 3). In every other feature a linear mapping was done taking as reference the minimum and maximum value of the feature in a set of experimenting instances. Not only complete instances are to be represented by the numerical vector, but also every intermediate state. A feature value could fall in the neighborhood out the range [0, 1]. That is why the hyperheuristic model considers labeling points outside this interval as detailed in [6].

In order to conform the new representation scheme for the instance state, we added an eighth feature regarding the fraction of the instance total items

Table 3. Representation **2** of the instance state

Feature	Description
1	Number of pieces.
2	Mean number of sides of remaining pieces.
3	Variance of the number of sides of remaining instance pieces.
4	Fraction of remaining pieces in the instance whose area is up to 1/4 of the object area.
5	Variance of the rectangularity of remaining pieces.
6	Variance of the width of the remaining pieces.
7	Variance of the area of the remaining pieces.
8	Fraction of the instance total items remaining.

that remain to be packed, so the GA learning process can have a sense of how advanced is the solution of a given instance (this is also done in [6], see Table 2). So, the representation scheme for the 2D irregular BPP which is the result of the proposed methodology is shown in Table 3. The number of features selected is the same for representation scheme 1 and 2 for a fair comparison.

Analyzing one of the most relevant features, *Mean number of sides of remaining pieces*, related with performance of heuristics BF+BLI and BFD+CA (see Table 1), we found out that the larger the mean number of sides the better idea to use heuristic BF+BLI and the worse idea to use heuristic BFD+CA. This is shown in Fig. 1. *Absolute* performance for all single heuristics is better as mean number of sides increases (correlation indexes between heuristic performance and mean number of sides go from 0.53 to 0.79 along the 40 heuristics). Mean number of sides goes from 3.2 to 4.1; so, it seems easier to place four-sided pieces than triangles. Nevertheless, heuristic BFD+CA decreases its *normalized* performance as mean number of sides increases. This means that, heuristic BFD+CA, although increases its absolute performance while mean number of sides increases (like all others heuristics), this performance does not improve as much as other heuristics do, making heuristic BFD+CA, in general, less adequate for solving instances with higher mean number of sides. This is what can be said when analyzing this feature alone. It is expected some interaction effect among several feature values, and this justifies the use of an evolutionary computation based approach for developing a set of rules for heuristic selection.

4 Experiments and Results

This section compares the quality of the hyper-heuristics generated when instances states are summarized employing representation schemes 1 and 2 (Tables 2 and 3). A total of 540 problem instances containing irregular pieces were randomly produced [6]. A problem instance with convex polygons from the literature named *Fu* [13] was added to the set. All pieces are convex polygons with a number of sides between 3 and 8 and the testbed includes 30 rectangular instances (details in [6]). Number of pieces per instance are in the range from 12 to 60. There is a variety of instance feature values; for example, there are instances whose pieces have an average size of 1/30 of the object, while other instances have huge pieces (averaging 1/3 of object each piece). Average pieces

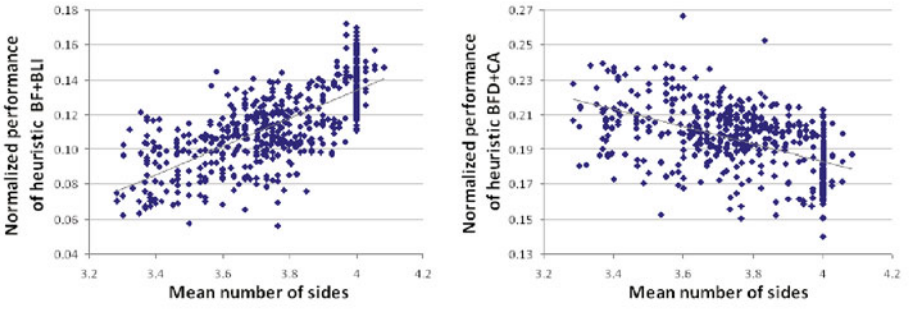


Fig. 1. Normalized performance of heuristics BF+BLI and BFD+CA related with instances mean number of sides

rectangularity goes from 0.35 to 1 along the 541 instances. Two experiments were conducted sorting the 541 available instances in two balanced training and testing sets and other two experiments swapped training and testing sets. Four experiments overall. Experiments were conducted with population of size 30, crossover probability of 1.0, mutation probability of 0.10, for 80 generations. For each of the four experiments with each representation scheme (1 and 2), two GA processes were run. For each complete run the best two individuals of all over the process were selected as hyper-heuristics. That is, four hyper-heuristics were obtained for every representation scheme in each of the four experiments. Overall, sixteen hyper-heuristics were employed to measure the effectiveness of each representation scheme.

Each hyper-heuristic generated was used to solve the training and testing sets of instances of the experiment where it comes from. Results are shown in Table 4. Figures in cells indicate the percentage of problems solved by hyper-heuristics with representation 2 that employs a particular number of extra objects (left column) when compared against results provided by the previous representation scheme. For example, when solving the training set instances of experiment II, hyper-heuristics developed using representation scheme 2 required at least 4 objects less in 10.0% of instances when comparing with hyper-heuristics developed using representation scheme 1. In previous experimentations, it has been confirmed that solving an instance with a hyper-heuristic is faster than solving it with each of the single heuristics and then choosing the best result, as well as that the method to create hyper-heuristics and the hyper-heuristics themselves are efficient with respect to the number of objects used [6].

Table 5 summarizes the results for all experiments. Comparing the average number of objects employed, the proposed new representation scheme uses 0.446 and 0.217 less objects for the training and testing sets respectively, averaging over results of the 16 hyper-heuristics. In Table 5 most numbers are negative, meaning that representation 2 deliver in average less objects than representation 1. We found out that the number of objects employed by hyper-heuristics is statistically lower with representation scheme 2 than with representation scheme 1 (p-value < 0.001). The sample size was 541 instances.

Table 4. Number of extra objects delivered by hyper-heuristics using representation scheme 2 compared with those using representation scheme 1. Experiments I through IV.

	Training Sets				Testing Sets			
	I	II	III	IV	I	II	III	IV
≤ -4		10.0						
-3	0.4	1.1	0.4	2.6			1.1	1.5
-2	3.3	1.1	7.0	3.0	0.4	1.1	8.1	3.0
-1	14.0	45.6	40.2	11.9	11.9	25.8	38.5	10.7
0	72.3	41.5	52.0	80.4	74.4	66.1	51.5	81.9
1	10.0	0.7	0.4	2.2	13.0	6.3	0.7	3.0
2					0.4	0.7		

Table 5. Average number of extra objects obtained by hyper-heuristics using representation scheme 2 compared with those using representation scheme 1

Exp.	Training	Testing
I	-0.104	0.037
II	-0.959	-0.203
III	-0.525	-0.534
IV	-0.196	-0.166
Avrg	-0.446	-0.217

5 Conclusions and Further Research

We have proposed a way to define a numerical vector representation for an instance of a given problem. Applying this methodology to the 2D irregular BPP we found a group of features to conform the representation scheme. We showed that building hyper-heuristics with the proposed representation scheme produces considerable better results than employing the representation scheme used in [6]. The developed methodology could be applied to any optimization approach that needs to represent instances states through numerical vectors; for example, the Case-based reasoning approach [8] or hyper-heuristics approaches where a number of features is used to categorize instances for choosing an adequate heuristic [2], [3], [4], [5], [7]. For future works, we will test the proposed methodology to enhance performance in some of these other problems and approaches.

Acknowledgments. This research was supported in part by ITESM under the Research Chair CAT-144 and the CONACYT Project under grant 99695.

References

1. Ross, P.: Hyper-heuristics. In: Burke, E.K., Kendall, G. (eds.) Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, pp. 529–556. Springer, New York (2005)
2. Ross, P., Schulenburg, S., Marín-Blázquez, J.G., Hart, E.: Hyper-heuristics: Learning to Combine Simple Heuristics in Bin-Packing Problems. In: GECCO 2002, pp. 942–948. Morgan Kaufmann Publishers Inc., San Francisco (2002)

3. Ross, P., Marín-Blázquez, J.G., Schulenburg, S., Hart, E.: Learning a Procedure that Can Solve Hard Bin-Packing Problems: A New GA-based Approach to Hyper-heuristics. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 1295–1306. Springer, Heidelberg (2002)
4. 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: GECCO 2006, Seattle, Washington, USA, pp. 591–598. ACM Press, New York (2006)
5. Terashima-Marín, H., Flores-Álvarez, E.J., Ross, P.: Hyper-heuristics and Classifier Systems for Solving 2D-regular Cutting Stock Problems. In: GECCO 2005, Washington DC, USA, pp. 637–643. ACM Press, New York (2005)
6. Terashima-Marín, H., Ross, P., Farías-Zárate, C.J., López-Camacho, E., Valenzuela-Rendón, M.: Generalized Hyper-heuristics for Solving 2D Regular and Irregular Packing Problems. Annals of Operations Research (2008)
7. Terashima-Marín, H., Ortiz-Bayliss, J.C., Ross, P., Valenzuela-Rendón, M.: Hyper-heuristics for the Dynamic Variable Ordering in Constraint Satisfaction Problems. In: GECCO 2008, Atlanta, GA, USA, pp. 571–578. ACM Press, New York (2008)
8. Petrovic, S., Qu, R.: Case-Based Reasoning as a Heuristic Selector in Hyper-Heuristic for Course Timetabling Problems. In: The 6th International Conference on Knowdelge-Based Intelligent Information Engineering Systems and Applied Technologies (KES), pp. 336–340 (2002)
9. Wäscher, G., Haubner, H., Schumann, H.: An Improved Typology of Cutting and Packing Problems. European Journal of Operational Research 3, 1109–1130 (2006)
10. Dyckhoff, H.: A typology of Cutting and Packing Problems. European Journal of Operational Research 44, 145–159 (1990)
11. Arthur, D., Vassilvitskii, S.: How Slow is the k-means Method? In: The 22nd Annual Symposium on Computational Geometry, Sedona, Arizona, USA, pp. 144–153. ACM Press, New York (2006)
12. Glonek, G.F.V., McCullagh, P.: Multivariate Logistic Models. Journal of the Royal Statistical Society. Series B (Methodological) 57, 533–546 (1995)
13. Fujita, K., Akagji, S., Kirokawa, N.: Hybrid Approach for Optimal Nesting Using a Genetic Algorithm and a Local Minimisation Algorithm. In: The 19th Annual ASME Design Automation Conference, Part 1 (of 2), Albuquerque, NM, USA, pp. 477–484 (1993)

Digital Web Library of a Website with Document Clustering

Isabel Mahecha-Nieto and Elizabeth León

Universidad Nacional de Colombia
Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
iamahechan@unal.edu.co, eleonguz@unal.edu.co

Abstract. Digital libraries allow organizing, classifying and publishing collections of electronic contents that are available in computers or networks. Also, digital libraries are easy to use and configure and they offer a user interface with access to fast searching and browsing over a repository of documents using a graphical interface. This article presents a digital library prototype for retrieving, indexing and clustering documents published on a website. The website may include unstructured, semi-structured and structured documents such as: web pages, scientific papers, news and documents in several formats that contain essentially text. The proposed prototype includes a clustering process that uses a conceptual algorithm and an *a priori* process of cluster labeling. Preliminary results correspond to tests made with different sets of documents published in a real website.

Keywords: Digital Web Libraries, Document Clustering, Information Retrieval, Websites.

1 Introduction

Nowadays, huge repositories of documents are important assets for organizations since they store information and knowledge required for developing daily activities inside them. In order to organize, classify and publish documents, organizations use web portals, search engines, FTP (File Transfer Protocol) servers, shared folders and, most recently, digital libraries.

These tools have had an important development since the 1990's, but the use of these systems is still limited to a few repositories [1]. Digital libraries have several advantages compared to other types of information retrieval systems because they integrate components for storing, classifying, retrieving and searching for documents. But a major drawback with the implementation of these systems is the effort needed to adapt them to the specific characteristics of an institution, company or organization [1].

In order to meet the basic expectations of users, digital libraries must provide some basic characteristics such as: usability in their user interface and configuration, user interface with access to fast searching and browsing over a repository of documents using a graphical interface designed with ergonomic features [1].

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 214–223, 2010.

This paper presents a digital library system adapted to an organization website. This system integrates an information retrieval subsystem with components for crawling, fetching and indexing. Also, this system has a component for automatic preprocessing and unsupervised clustering of structured, unstructured and semi-structured documents.

The proposed digital library has an user interface that is based on a tree structure for browsing over labeled sets of related documents. This kind of browsing scheme is similar to that used by search engines such as Yippy[1](formerly Clusty) and Carrot2[2] that include clustering search results.

Retrieving and clustering the documents of an institution with the digital library proposed in this work provides a useful way to access the information stored into its website, even if an user does not know the particular links or URI of web pages or contents. Also, the system was designed with a manager interface for configuring required parameters of each component. This interface can update the system by repeating the process of retrieving and classifying when are uploaded new documents to the website.

This paper encompasses other four sections. Section 2 gives a brief background on digital libraries. Section 3 describes the proposed system architecture. Section 4 describes the prototype of the proposed system. Section 5 concludes with the main findings of the study.

2 Digital Libraries

2.1 Concept and Features

A digital library is defined as a collection of electronic contents that is available in computers or networks. According to Levy and Marshall [2], a digital library is an integration of documents, technology and work. This definition shows the importance of human labor for successfully create and improve this kind of

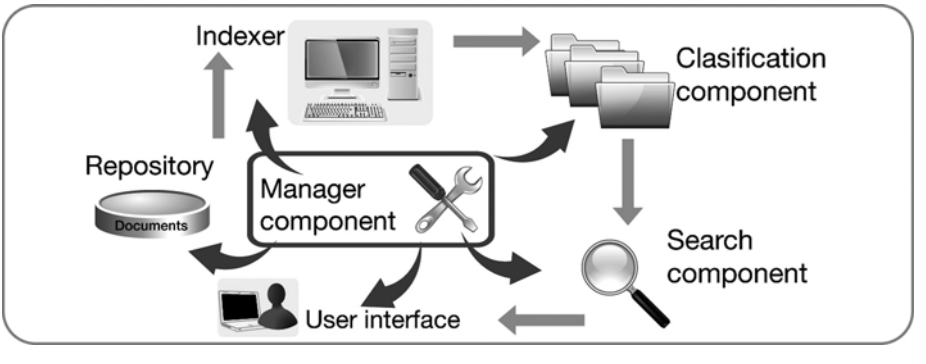


Fig. 1. Architecture of a Digital Library

[1] Yippy http://search.yippy.com/
[2] Carrot2 http://search.carrot2.org/stable/search

systems. Lesk [3] analyzed the advances made in our understanding on the concept of digital library and he concluded that advances on this matter are divided into the comprehension of the users' requirements and the use of the resources that digital libraries provide.

Now, we can define a digital library as an extended system of information retrieval[3] [4] that contains collections of documents stored in digital formats, databases, knowledge bases and the World Wide Web (WWW). Figure 1 shows an overview of a digital library where relationships between its different components are illustrated.

2.2 Digital Libraries Software Systems

These systems usually have basic functions of information retrieval and their collections of documents are built under the supervision of an expert who selects the name for such collections and then classifies the documents with a predefined strategy. Some of the most famous digital libraries software systems are: Greenstone [1], Fedora [5], DSpace [6], EPrints [7] and OpenDLib [8].

Other approaches for document classification and clustering are based on data mining models such as MARIAN [9], BT Digital Library [10], Somelib [11] and Hypia System [12]. Moreover, research on semantic web has proposed other systems supported on ontologies, meta-data and taxonomies, such as the system proposed on [13] that is described as a methodology for building an automatic taxonomy inside a digital library.

All these approaches try to solve the problem of document classification, but the main difficulty is that an acceptable classification process is not automatic but rather needs of an expert to refine sets of documents's labels and relationships among them. In general, digital libraries use structured contents such as scientific publications, but in other contexts like websites, almost all content is unstructured but they represent is a huge asset for organizations. This work presents an approach to integrate a process for clustering unstructured documents inside a digital library.

3 Proposed System Architecture

This section presents a description of the system as well as an overview of the complete prototype, its components and operation process. Figure 2 illustrates the system's architecture and processing flow. Subsection 3.1 describes the information retrieval system that is used to perform three tasks: fetching the URI of the documents stored in the web portal, crawling documents and indexing their text content. Subsection 3.2 explains the web mining process used to classify the documents stored in the web portal. Finally, subsection 3.3 describes the user interface.

[3] Extended system of information retrieval is a concept applied to define a digital library, since this type of system can be formed by the fusion of multiple sets of documents and different media content [4].

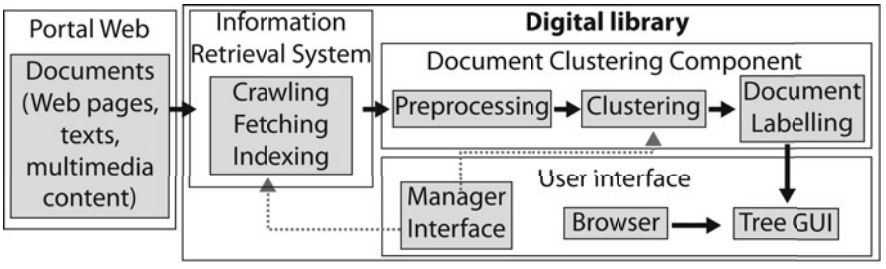


Fig. 2. System's Architecture

3.1 Information Retrieval System

The information retrieval system performs two basic tasks: crawling and indexing. This system includes some of the components of Nutch, which is an open source web-search engine hosted at The Apache Software Foundation[4]. It is built on one of the most widely used indexing libraries, Lucene[5], and has been added web specific components such as a crawler, a link-graph database, parsers for plain text, HTML, RTF, PDF, JavaScript and Microsoft Office [14]. The crawling and indexing components of Nutch were selected according to the methodology described in [15]. Figure 3 shows the information retrieval process. Input data comprise all documents published on a website.

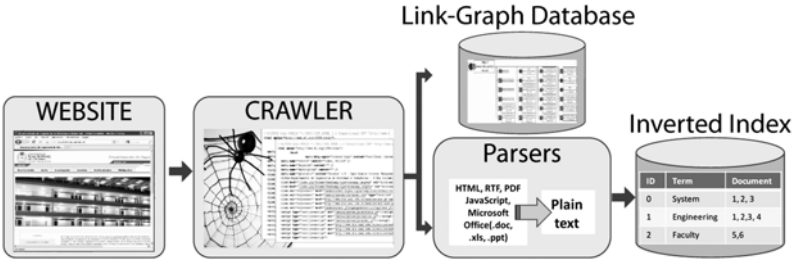


Fig. 3. Information Retrieval System

Crawling. This task is developed using three components: a web database, fetchers and parsers. A web database contains a repository of the URIs of the pages that were downloaded, which is called a fetch-list [16]. After all fetch-lists have been updated in the web database, fetchers read the fetch-lists and start downloading web page's contents along with stored documents [16]. Then, the system selects an appropriate parser for the document and extracts all its text content, the title and URI. All these contents are stored in an index.

[4] Nutch http://lucene.apache.org/nutch/about.html
[5] Lucene http://lucene.apache.org/java/docs/

Indexing. The indexing process uses the contents and documents of the web pages to build an inverted index. The index stores all the terms present in the contents and their frequency in each document. While the indexing process is performed, the set of documents are transformed into smaller sets of index segments, so with this strategy it is possible to distribute them into several search processes for a faster scaling of the indexing process [16].

3.2 Document Clustering Component

Document clustering is performed with Lingo, which is a conceptual clustering algorithm [17]. Lingo is based on singular value decomposition (SVD) of a term-frequency matrix and an *a priori* process of cluster labeling. This algorithm uses an innovative approach compared to other clustering algorithms [18]. Traditional clustering algorithms used the following process: first, the content of the cluster is discovered and then the labels for the clusters are specified according to this content [17]. The problem with this approach is that it was difficult to find a plain human explanation to why a set of documents were in the same cluster. Lingo proposed another scheme to solve the problem of the classical algorithm by finding first a set of cluster labels that are humanly understandable and then assigning documents to each labeled cluster [17].

Lingo has five sequential phases: preprocessing, frequent phrase extraction, cluster label induction, cluster content discovery and final cluster formation [17]. For the prototype presented in this paper, the Lingo algorithm was adapted to process the entire content of document sets and not only the search result snippets of a search engine, which is the original idea of this algorithm. This change was introduced because the experimentation system is focused on the entire contents of a set of documents and one of the goals was to identify a classification strategy for that set using the complete content of the documents.

Preprocessing. The content, URI and title of each document comprise the input to be fed into the Lingo algorithm. During document preprocessing, non-letter characters (except for sentence boundaries) are filtered. Stop words are detected by searching for occurrences of stop words previously identified in different languages and stored in lists. The language of each document is identified by counting the greater number of stop words for a specific language [17]. Also, a process of stemming is executed using Porter's algorithm for English documents and the Snowball Library[6] for other languages such as Spanish and French.

Feature extraction. This step consists in discovering candidate labels for the clusters. Labels are phrases and frequent terms that have a potential meaning or human-readable topic descriptions for the content of a cluster [17]. For more details about the algorithm used to identify complete phrases see [17]. Candidate phrases should have the following characteristics: 1. They should belong to the set of input documents at least by a given minimum of times, 2. They should not cross sentence boundaries and 3. They should not begin nor end with a stop word.

[6] Snowball http://snowball.tartarus.org/

Cluster label induction. This phase has the goal of finding meaningful descriptions of each cluster using SVD decomposition. It includes four steps: 1. Term-document matrix building, 2. Abstract concept discovery, 3. Phrase matching and 4. Label selection. The term-document matrix is built by calculating the tfidf weight (term frequencyinverse document frequency) of each term [19]. Final labels for the clusters are selected by comparing all labels and discarding the ones that are too similar to each other. Complete details of this process are also described in [17].

Cluster content discovery. The clusters are formed using the widely used Vector Space Model (VSM) [19], while Lingo is used to match the input documents to every cluster label. Documents are not assigned to any cluster ending in an artificial group called "other topics" [17]. In the prototype described in this paper, all documents stored in the website are matched to labels, so the performance is worse but it is carried out successfully because the process is not performed online.

Cluster formation. Finally, clusters are formed by calculating the relevance of each document inside its own cluster. This is calculated by multiplying each label score and the number of snippets or documents in the cluster [17].

3.3 User Interface

The system's user interface has two parts: a graphic interface and a manager interface. The graphic interface consists of a tree diagram where nodes correspond to cluster names (labels) and branches correspond to the documents' title and URI. The browsing process allows users to reduce and expand the tree diagram; this ergonomic feature results in a more comfortable sight of the interface. The manager interface allows configuring the parameters for each of the system's components in order to run the information retrieval system functions and the clustering algorithm periodically.

4 Prototype of the Proposed System

The proposed system was implemented as a prototype and written in Java language. This language was selected because Nutch and Lingo are also written in Java. Moreover, the use of a Java framework enables good options to implement web applications. The prototype was tested with a real website of an academic institution.

4.1 Input Set of Documents

The chosen website was from the Department of Computer System Engineering and Industrial Engineering of the National University of Colombia[7]. Most of

[7] Department of Computer System Engineering and Industrial Engineering http://www.disi.unal.edu.co/

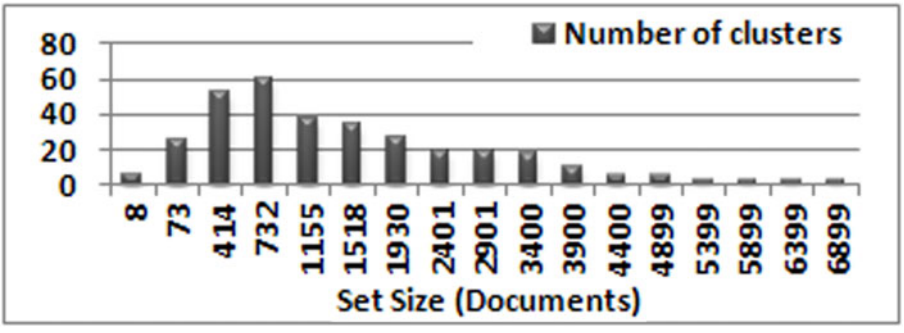


Fig. 4. Number of Clusters

the documents in this website are written in Spanish, while fewer are written in English. Documents published in the website were indexed in different test sets in order to examine the behavior of the system with sets of increasing sizes, obtaining seventeen sets. Fig. 4 shows the relationships between the number of documents and the number of clusters. The best sets of clusters were found in sets between 733 and 2401 documents.

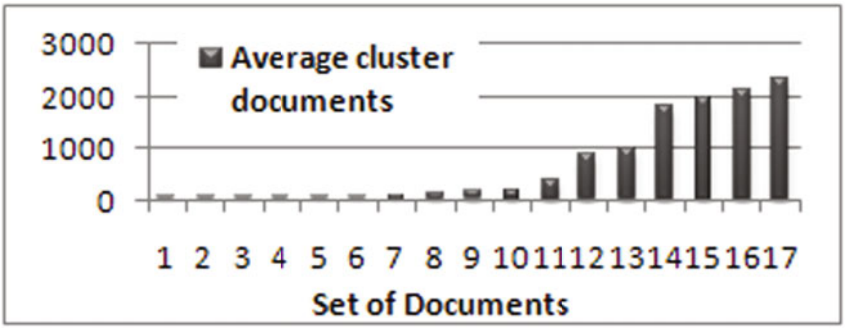
4.2 Prototype Parameters

Crawling and Indexing Parameters

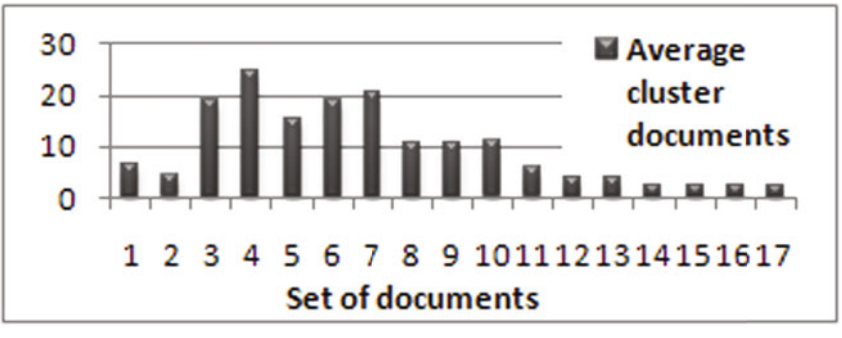
- Depth: This parameter refers to how far the crawler goes. In this case, different measures for this parameter were used: 2, 3 and 10 for a full crawl of the website.
- Number of pages per level: This is the limit for the number of pages fetched at each level of the web graph. Tests were done with six amounts of documents: 10, 50, 100, 200, 500 and 1,000.

Clustering Algorithm Parameters. A complete description of each parameter of the Lingo algorithm can be found in [17]. The clustering algorithm was tested using different values for the following parameters:

1. Candidate Label Threshold: It is the maximum number of candidate cluster labels. Values range between 0.7 and 0.9. The best results are obtained with 0.75.
2. Document Assignment Threshold: It is the similarity threshold that a document weight must exceed in order to be added to a cluster. Values range between 0.1 and 0.3. The best results are obtained with 0.2.
3. Term Frequency Threshold: It is the number of times that a term must appear in the input documents in order to be included in the term-document matrix. Values range 2-5. The best results are obtained with 3.
4. Label Similarity Threshold: It is the acceptable level of similarity using cosine distance between clusters. Values ranged between 0.2 and 0.5. The best results are obtained with 0.3.



(a) With Cluster "Other Topics"



(b) Without Cluster "Other Topics"

Fig. 5. Average Number of Documents per Cluster

4.3 Analysis of the Results

Cluster composition. Cluster composition was analyzed based on the average number of documents per cluster. Fig. 5 shows a comparison of average, figure a) shows the average including all clusters built in each set, and figure b) shows the average without including the cluster "other topics". This comparison was done because there was an algorithm trend to cluster many documents within the cluster "other topics." The best cluster compositions were found in sets with sizes ranging between 414 and 1,930 documents, because the average number or documents per cluster ranges between 15 and 20. In sets with more than 1,930 documents the cluster "other topics" is significantly large; this means that the clustering process does not find enough clusters in big sets of documents.

Cluster labels. The results obtained in this work show that appropriate labels are identified for about 80% of the clusters in sets with a size ranging between 4 and about 2,401 documents. Moreover, the clusters have a match with the documents contained in such clusters. A formal validation study is being carried out by experts and the complete results will be published in a future paper. Figure 6 shows an example of the results obtained. It presents a tree structure of labels and the content of two of the clusters.

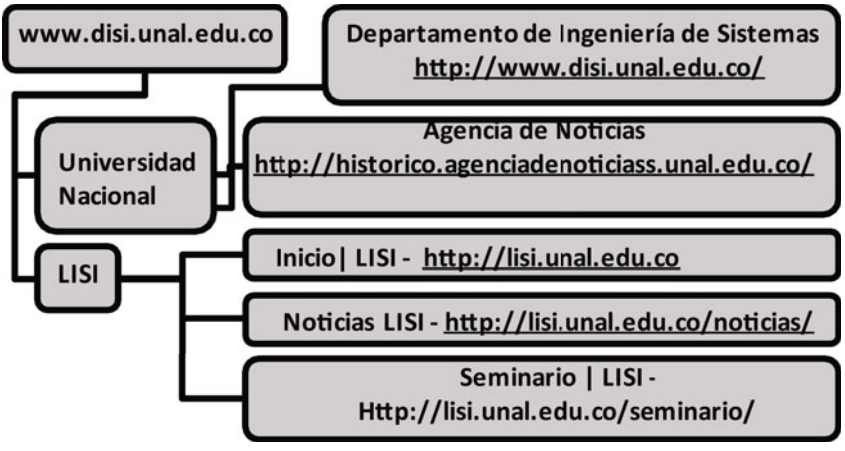


Fig. 6. Example of obtained results

5 Conclusions and Future Work

The integration of a conceptual clustering algorithm with an *a priori* process of cluster labeling in an algorithm like Lingo works efficiently for clustering search results and a complete set of text documents with different kinds of structures (structured, unstructured, semi-structured). Tests show that appropriate labels are assigned to clusters with small or medium document size sets (written in Spanish and English). Given that Lingo uses the singular value decomposition (SVD) method, it needs of a machine with enough RAM to maintain the document sets in order to perform successfully. Since some phases of document clustering are performed off-line, the time required to run the Lingo algorithm is not critical. One of the improvements of the proposed system could be the inclusion of a hierarchical methodology to show other relationships inside the contents of the clusters.

References

1. Witten, I.H., Boddie, S.J., Bainbridge, D., McNab, R.J.: Greenstone: a comprehensive open-source digital library software system. In: Proceedings of the Fifth ACM Conference on Digital Libraries, San Antonio, Texas, United States, pp. 113–121. ACM, New York (2000)
2. Levy, D., Marshall, C.: Going digital: A look at assumptions underlying digital libraries. Communications of the ACM 38, 77–84 (1995)
3. Lesk, M.: Understanding Digital Libraries, 2nd edn. Morgan Kaufmann, New Jersey (2004)
4. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval. Addison-Wesley, Harlow (1999)
5. Lagoze, C., Payette, S., Shin, E., Wilper, C.: Fedora: an architecture for complex objects and their relationships. International Journal on Digital Libraries 6(2), 124–138 (2006), http://dx.doi.org/10.1007/s00799-005-0130-3

6. Tansley, R., Bass, M., Smith, M.: DSpace as an open archival information system: Current status and future directions. In: Koch, T., Sølvberg, I.T. (eds.) ECDL 2003. LNCS, vol. 2769, pp. 446–460. Springer, Heidelberg (2003)
7. Gumpenberger, C.: The eprints story: Southampton as the cradle of institutional self-archiving. GMS Medizin - Bibliothek - Information 9, 1–6 (2009)
8. Castelli, D., Pagano, P.: OpenDLib: a digital library service system. In: Agosti, M., Thanos, C. (eds.) ECDL 2002. LNCS, vol. 2458, pp. 327–340. Springer, Heidelberg (2002)
9. Gonçalves, M., France, R., Fox, E.: MARIAN: Flexible interoperability for federated digital libraries. In: Constantopoulos, P., Sølvberg, I.T. (eds.) ECDL 2001. LNCS, vol. 2163, pp. 173–186. Springer, Heidelberg (2001)
10. Bloehdorn, S., Cimiano, P., Duke, A., Haase, P., Heizmann, J., Thurlow, I., Völker, J.: Ontology-based question answering for digital libraries. In: Kovács, L., Fuhr, N., Meghini, C. (eds.) ECDL 2007. LNCS, vol. 4675, pp. 14–25. Springer, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-74851-9_2
11. Rauber, A., Merkl, D.: Text mining in the SOMLib digital library system: The representation of topics and genres. Applied Intelligence 18(3), 271–293 (2003), http://dx.doi.org/10.1023/A:1023297920966
12. Finn, A., Kushmerick, N., Smyth, B.: Fact or fiction: Content classification for digital libraries. In: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries (2001)
13. Sanchez, D., Moreno, A.: Bringing taxonomic structure to large digital libraries. International Journal of Metadata, Semantics and Ontologies 2(2), 112–122 (2007), http://dx.doi.org/10.1504/IJMSO.2007.016805
14. Software Foundation, T.A.: Nutch. Technical report, The Apache Software Foundation (2007), http://wiki.apache.org/nutch/
15. Mahecha-Nieto, I., León Guzmán, E.: An exploratory study of open source search engines: Evaluation of supportability, usability, functionality and performance. In: Quinto Congreso Colombiano de Computación 2010 (2010)
16. Cafarella, M., Cutting, D.: Building Nutch: Open Source Search. Queue 2(2), 54–61 (2004), http://doi.acm.org/101145/g88392.g88408
17. Osiriski, S., Stefanowski, J., Weiss, D.: Lingo: Search results clustering algorithm based on singular value decomposition. In: Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM 2004 Conference Held in Zakopane, Poland, May 17-20, pp. 359–368. Springer, Heidelberg (2004)
18. Osiriski, S., Weiss, D.: Conceptual clustering using lingo algorithm: Evaluation on open directory project data. In: Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM 2004 Conference Held in Zakopane, Poland, May 17-20, p. 369. Springer, Heidelberg (2004)
19. Salton, G.: Automatic text processing: the transformation, analysis and retrieval of information by computer. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)

Learning to Discover Faulty Spots in cDNA Microarrays

Mónica G. Larese[1,2,⋆], Pablo M. Granitto[1], and Juan C. Gómez[1,2]

[1] CIFASIS, French Argentine International Center for Information and Systems Sciences, UPCAM (France) / UNR-CONICET (Argentina)
Bv. 27 de Febrero 210 Bis, 2000, Rosario, Argentina
[2] Lab. for System Dynamics and Signal Proc., FCEIA, Univ. Nacional de Rosario
Riobamba 245 Bis, 2000, Rosario, Argentina
{larese,granitto}@cifasis-conicet.gov.ar,
jcgomez@fceia.unr.edu.ar
http://www.cifasis-conicet.gov.ar/

Abstract. Gene expression ratios obtained from microarray images are strongly affected by the algorithms used to process them as well as by the quality of the images. Hundreds of spots often suffer from quality problems caused by the manufacturing process and many must be discarded because of lack of reliability. Recently, several computational models have been proposed in the literature to identify defective spots, including the powerful Support Vector Machines (SVMs). In this paper we propose to use different strategies based on aggregation methods to classify the spots according to their quality. On one hand we apply an ensemble of classifiers, in particular three boosting methods, namely Discrete, Real and Gentle AdaBoost. As we use a public dataset which includes the subjective labeling criteria of three human experts, we also evaluate different ways of modeling consensus between the experts. We show that for this problem ensembles achieve improved classification accuracies over alternative state-of-the-art methods.

Keywords: cDNA microarray images, consensus-based prediction, ensemble algorithms, spot quality control, classification of spots.

1 Introduction

Complementary DNA (cDNA) microarray images are a high throughput technology developed in the last decades aimed at comparing the gene expression levels of thousands of genes simultaneously [1,2,3,4]. Each one of the thousands of microscopic spots available in a microarray image corresponds to a certain gene under study. The ratio between the intensities corresponding to the target and to the probe samples measures the hybridization level of the gene. Ideally, all the spots should be circular, with uniform spatial distributions of intensities and

⋆ Author to whom all correspondence should be addressed. Work supported by ANPCyT under grant PICT 2226 and Universidad Nacional de Rosario.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 224–233, 2010.

good contrast. However, deficiencies in the manufacturing and hybridization processes produce hundreds of faulty spots which must be discarded in early steps of the analysis in order to avoid unreliable measurements.

Since the appearance of microarray technology, hundreds of works have been proposed to automatically process them. Most of the works existing in the literature concentrate on developing new algorithms to locate and segment the spots [5,6,7,8], or to perform data mining tasks in order to analyze the huge amount of information that they provide [9,10,11,12]. However, a few papers deal with the very important task of detecting faulty spots and filtering them out. Existing software packages, such as GenePix Pro [13] and Scanalyze [14], let the user flag bad spots manually. Several morphological and statistical additional features, such as spot size, spot circularity or signal to noise ratio, are also automatically computed to describe the spots. These features are then expected to be combined and thresholded in order to discard spots with undesirable characteristics.

In the last years only a few works proposed to learn a computational model of the spots able to discriminate those with bad quality. Ruosaari and Hollmen [15] extract spatial spot features and use them to train a Bayesian binary classifier which separates the spots according to their good/bad quality. Hautaniemi *et al.* [16] propose to fit a 2D Gaussian distribution to every spot to extract several features. Then they apply a quality control strategy based on Bayesian networks to perform classification. The structure of the network, describing the relations between the model components, must be described first. In Bicego *et al.* [17] the recent and powerful Support Vector Machines (SVMs)[18] are proposed to separate good quality from bad quality spots using the same features as in Hautaniemi *et al.* [16]. This method shows to improve the performance, and has the advantage that it does not require *a priori* knowledge about the data.

In this paper we propose to improve classification by learning a model of the spots based on aggregation methods, specifically ensembles and consensus theory. For this purpose we analyze real spots from publicly available microarray images which have been manually labeled by three human experts. To the best of our knowledge, aggregation methods have not been proposed for quality control of the spots yet, even though ensembles have shown to be very competitive against the best state-of-the-art learning algorithms, *i.e.* SVMs, achieving similar or even better performances (see e.g. Liu *et al.* [19]). In the paper by Baluja *et al.* [20] some ensembles also show to be faster than SVMs.

Ensemble methods [21] are machine learning algorithms developed in the last decades which leverage the power of multiple learners and combine their predictions in order to achieve better accuracy and more robustness than any of the single learners acting individually. The learners should be complementary to one another to take advantage of the method, because if they always agree there would not be any improvements over using the individual learners.

On one hand we propose to solve the binary classification problem (to separate good from bad quality spots) by using Boosting, a powerful representative of ensemble methods. Boosting is an iterative algorithm based on the idea

that if several "weak" classifiers (simple classification rules which provide misclassification errors slightly better than chance) are combined into an ensemble, the result will be a "strong" classifier with a highly improved performance [22,23]. Perhaps the most common and simplest version of boosting is the AdaBoost algorithm [24], also called Discrete AdaBoost. Several variants have been proposed since Discrete AdaBoost appearance, including Real and Gentle AdaBoost [25].

The three human experts that evaluated the spot images only agree about the labeling criteria in less than 50% of the spots. This happens because, even though the individuals have worked with microarrays for many years, they have different beliefs, backgrounds and experiences. The basic approach to deal with this problem is to simply ignore it: the dataset is reduced to the 155 spots on which the three experts have a complete agreement. Unfortunately, by doing so one is probably disregarding useful information about what is a good or bad spot for each individual expert. In this work we also consider an alternative way to manage the problem of lack of agreement, which is the use of a consensus model. We propose to build a set of ensemble classifiers, each one modeling the opinion of one of the experts. Then we use the individual models as a committee of experts, in which each classifier has one vote and the majority opinion is considered as the output of the committee. This model has the potential to incorporate the complete contribution, experience and beliefs of each expert.

We compute the generalization errors for each method and compare our two proposed approaches to previously published results available in the literature. We show that one of our methods achieves improved classification accuracies.

The rest of the paper is organized as follows. In Section 2 we describe the dataset. We explain the extracted features and the implemented classification algorithms in Section 3. We show the experimental results in Section 4. Finally, we draw some conclusions in Section 5.

2 Dataset Description

We use a publicly available dataset[1] which consists of spots extracted from two different microarray images [16]. One of the grids from these images is shown in Fig. 1. A total number of 320 spots (160 from each image) were divided by three human experts, which have several years of experience dealing with microarray experiments, into four quality categories: bad, close to bad, close to good and good. The labeling of the three experts exactly coincides in 155 spots. In order to perform binary classification, we grouped the previously mentioned four quality categories into good quality (by joining the sets "good" and "close to good") and bad quality (the union of sets "bad" and "close to bad"). Using these settings, we found 97 out of the 155 spots to have good quality and 58 bad quality. These same settings were used by Hautaniemi *et al.* [16] and Bicego *et al.* [17].

[1] The microarray images, labeling of the experts and additional information about the dataset can be downloaded from `http://www.cs.tut.fi/TICSP/SpotQuality/`

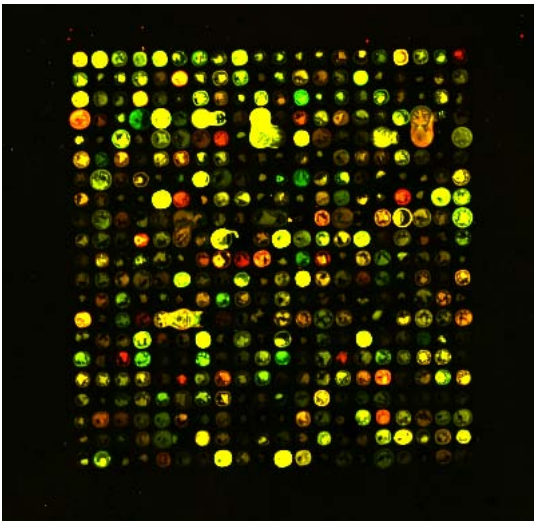


Fig. 1. One of the microarray grids containing the spots to be classified

In Fig. 2 we show two examples of spots. The spot in Fig. 2(a) has good quality, *i.e.*, it has a circular shape, with good contrast and uniform spatial intensity distribution, while the spot in Fig. 2(b) spreads into its neighbors, resulting in a bad quality spot.

3 Spots Classification

In this section we describe the set of features used to characterize the spots as well as the proposed classification algorithms.

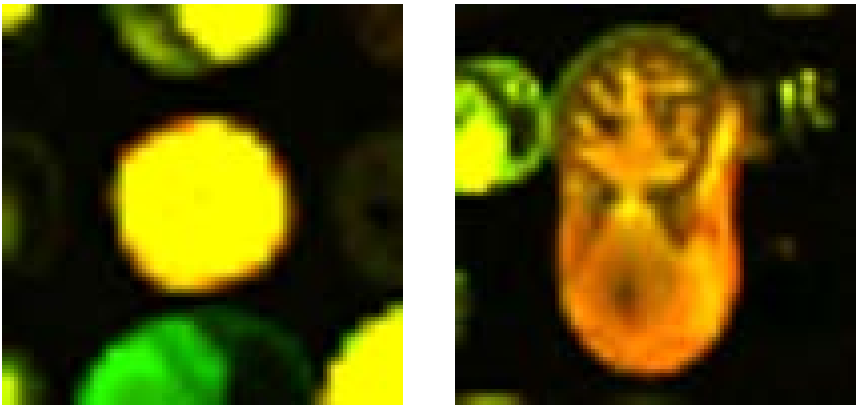
3.1 Set of Features

We computed 14 features per spot (7 for each channel corresponding to the probe and target samples), as proposed by Hautaniemi *et al.* [16]. We compute the seven features as follows.

First of all, we fit a 2D Gaussian surface to every spot using a standard non-linear least squares procedure. This function is the estimation of the spot intensity distribution across the spot pixels. The 2D function is defined as [16]

$$f(\mathbf{x}, A, B, \bar{\mathbf{x}}, \sigma_x, \sigma_y, \phi) = Ae^{-(\mathbf{x}-\bar{\mathbf{x}})^T \mathbf{S}(\mathbf{x}-\bar{\mathbf{x}})} + B \tag{1}$$

where $\mathbf{x} = [x, y]^T \in \mathbb{R}^2$ is the pixel coordinates vector, $\bar{\mathbf{x}} \in \mathbb{R}^2$ is the Gaussian mean, and $\mathbf{S} = R_\phi^T \mathrm{diag}\left(\sigma_x^{-2}, \sigma_y^{-2}\right) R_\phi$ is the inverse covariance matrix, where σ_x and σ_y are the x and y standard deviations, respectively, and R_ϕ is the rotation matrix with rotation angle ϕ. Parameters A and B are the foreground and background intensities of the spot, respectively.



(a) Good quality spot. (b) Bad quality spot.

Fig. 2. Examples of good and bad quality spots

We extract the seven features from this 2D Gaussian function in the following way (the reader is referred to Hautaniemi *et al.* [16] for more details on how to compute these features):

- Spot intensity: parameter A.
- Background intensity: parameter B.
- Alignment error: the distance between $\bar{\mathbf{x}}$ and the center of the spot bounding box.
- Roundness: the σ_x/σ_y rate.
- Spot size: the $\sigma_x\sigma_y$ product.
- Background noise: the root mean square error between the spot and the fitted 2D Gaussian function.
- Bleeding: the number of pixels of the spot which fall outside the fitted Gaussian.

The final feature vector is composed by the aforementioned continuous features. In contrast to previous works ([16] and [17]), we do not apply any discretization procedures nor feature selection algorithms on the set of features in order to perform binary classification, since it does not provide any improvements in our case.

3.2 Boosting

Boosting classifiers are based on the idea that if many "weak" classifiers (slightly better than chance) are combined into a "strong" classifier, the overall performance will be highly improved [22,23]. In this paper we consider three different boosting algorithms, namely Discrete, Real and Gentle AdaBoost, which we describe below.

Let $D = \{(\mathbf{x}_i, y_i)\}$, with $i = \{1, ..., n\}$, be a training dataset of n pairs of feature vectors $\mathbf{x}_i \in \mathbb{R}^p$ and class labels $y_i \in \{-1, 1\}$. Discrete AdaBoost [24] creates a sequence of weak classifiers ($f_m(\mathbf{x}_i)$) aimed at discriminating the training observations. Initially, all the observations are assigned a unique weight $w_{i,m}$. This distribution of weights is modified along with the $m = \{1, ..., M\}$ iterations (rounds), *i.e.*, observations which are badly classified (more difficult to learn) are given higher weights. The algorithm attempts to find an optimum classifier at each round. Each weak classifier is weighted according to its performance on the current distribution of weights on the observations. At the end, the final strong classifier $F(\mathbf{x}_i)$ is the linear combination of the weak classifiers, as follows

$$F(\mathbf{x}_i) = \text{sign}\left(\sum_{m=1}^{M} w_{i,m} f_m(\mathbf{x}_i)\right) \tag{2}$$

The output of Discrete AdaBoost at each iteration is a discrete value corresponding to the predicted class label for each observation. The efficiency of the algorithm may be improved by computing class probabilities instead of discrete labels. These class probabilities are then converted to the real scale and used to update the weight distribution for the observations at each iteration. This improved algorithm is named Real AdaBoost [25]. Both Discrete and Real AdaBoost minimize the expectation of the so-called "exponential loss", defined as $e^{-y_i F(\mathbf{x}_i)}$. Gentle AdaBoost [25] is an algorithm very similar to Real AdaBoost, but uses a sequence of Newton steps to optimize the expectation of the exponential loss. Even though the classification results are very similar for both methods, this feature makes Gentle AdaBoost numerically superior to Real AdaBoost.

3.3 Consensus of Experts

Consensus theory [26] combines multiple single probability distributions in order to build a unique predictive model from the opinion of several experts, assuming that the individual judgments are based on Bayesian decision theory.

The M probability distributions from each one of the M experts are combined using a so-called *consensus rule*. The consensus rule P_k for each pattern $\mathbf{x}_i$, with $i = \{1, ..., n\}$, is calculated for each one of the K classes. The simplest and most common consensus rule is the *Linear Opinion rule* (LOP), which is computed as a linear combination of the posterior probabilities P_j from each expert (Eq. 3), with $j = \{1, ..., M\}$, and α_j denoting the weights associated to each expert. The coefficients α_j are non-negative and $\sum_{j=1}^{M} \alpha_j = 1$. The simplest approach of the weighting scheme consists in assigning the same weights to all the experts.

$$P_k(\mathbf{x}_i) = \sum_{j=1}^{M} \alpha_j P_j(\mathbf{x}_i) \tag{3}$$

Pattern $\mathbf{x}_i$ is then classified to belong to class $\hat{k}_i$ (with associated label $\hat{y}_i$), which is the class with the highest probability, as below.

$$\hat{k}_i = \underset{k \in \{1, \cdots, K\}}{\text{argmax}} \; P_k(\mathbf{x}_i) \tag{4}$$

Table 1. Test errors for the different classifiers using LOOCV

Classification Algorithm	**Accuracy**
B-Course (subjective) [16]	96.8%
Pair-wise NB (subjective) [16]	95.5%
NB (subjective) [16]	95.5%
NB (uniform) [16]	94.8%
Decision Tree [16]	91.6%
Neural Networks [16]	90.3%
SVM (Linear)[17]	96.1%
SVM (Polynomial)[17]	94.2%
SVM (Gaussian RBF)[17]	97.4%
Consensus with Discrete AdaBoost (proposed approach)	96.1%
Consensus with Real AdaBoost (proposed approach)	97.4%
Consensus with Gentle AdaBoost (proposed approach)	96.1%
Discrete AdaBoost (proposed approach)	96.8%
Real AdaBoost (proposed approach)	97.4%
Gentle AdaBoost (proposed approach)	**98.1**%

4 Experimental Results

We implemented the three boosting algorithms, namely Discrete, Real and Gentle AdaBoost, using stumps as weak learners and 1000 rounds. We applied them to classify the 155 spots having the three experts' agreement about quality, and used leave-one-out cross-validation (LOOCV) to assess the performance of each model. We closely followed the same methodology used in previous works [16,17] in order to fairly compare our generalization errors to those obtained by these authors.

We also evaluated the use of a consensus of experts. For this purpose we modeled the individual criterion of each subject via the three boosting algorithms considered before, namely Discrete, Real and Gentle AdaBoost. We performed LOOCV on the total number of 320 spots available in the dataset. However, since the true labeling is established from the agreement of the three experts, the test error is computed only for the same 155 spots as before. In this way we can have a fair comparison of both strategies, using exactly the same test set to evaluate them.

We show in Table 1 the generalization errors for different classification algorithms. Comparing first the three boosting strategies, Gentle AdaBoost (highlighted in gray) achieves the best accuracy, with only 3 mis-classified spots out of the 155. Real AdaBoost follows with 4 errors, closing Discrete AdaBoost with 5 errors. The best previous method for this dataset is SVM with Gaussian RBF kernel [17], mis-classifying 4 spots. Two of the considered boosting strategies show equal or better performance than all previously published results.

In the same table we included the results obtained with the new consensus strategy combined with the same three boosting methods. The combination of

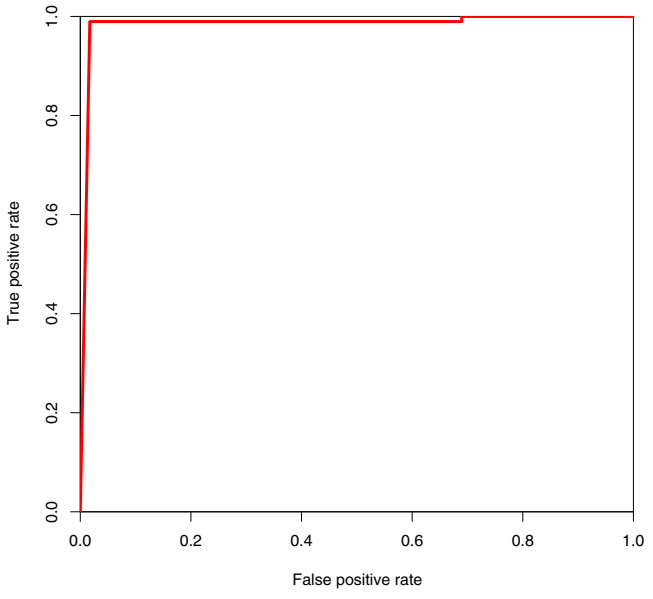


Fig. 3. ROC curve for Gentle AdaBoost classifier

consensus with Real AdaBoost shows the best result with 4 errors, but the two other combinations produced slightly lower results with 6 mis-classified spots. Overall, it seems that the consensus strategy is not able to outperform the more traditional methods.

We also computed the Receiver-Operating-Characteristic (ROC) curve for the classifier showing the best performance, *i.e.* Gentle AdaBoost. We show the ROC curve in Fig. 3, where the relation between the true positive rate (TPR) and the false positive rate (FPR) is depicted. The area under the ROC curve is equal to 0.984. It is interesting to notice the sharp shape of the curve. This is a consequence of the use of boosting strategies, which have a strong tendency to produce extreme probability estimations (almost always very close to 1 for the winning class).

5 Concluding Remarks

In this paper we propose to model the quality of spots from microarray images by means of aggregation methods. We considered specifically two different kinds of algorithms, namely ensemble and consensus theory based methods. The ensembles that we implemented correspond to Discrete, Real and Gentle AdaBoost. For the case of consensus theory, we proposed to model the criterion of each expert separately and then to combine the three experts' beliefs into a single predictive model.

Our experimental results show that Gentle AdaBoost obtains the best accuracy, even improving all the methods already proposed in the literature for the same dataset. On the other hand, the use of the consensus model, even though it has the potential to use more information than the traditional strategy, was not able to improve previous results on this problem.

Currently, we are working on the application of other ensembles to this problem. We are also considering the detection of specific defects, instead of a hard good/bad decision, which may help identify the degree of reliability of faulty spots.

References

1. Schena, M., Shalon, D., Davis, R.W., Brown, P.O.: Quantitative monitoring of gene expression patterns with a complementary cDNA microarray. Science 270, 467–470 (1995)
2. Lashkari, D.A., DeRisi, J.L., McCusker, J.H., Namath, A.F., Gentile, C., Hwang, S.Y., et al.: Yeast microarrays for genome wide parallel genetic and gene expression analysis. Proc. Natl. Acad. Sci. USA 94(24), 13057–13062 (1997)
3. Eisen, M.B., Brown, P.O.: DNA arrays for analysis of gene expression. Meth. Enzymol. 303, 179–205 (1999)
4. Alizadeh, A.A., Eisen, M.B., Davis, E.E., Ma, C., Lossos, I.S., Rosenwald, A., et al.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403(6769), 503–511 (2000)
5. Bengtsson, A., Bengtsson, H.: Microarray image analysis: background estimation using quantile and morphological filters. BMC Bioinformatics 7(96), 1–15 (2006)
6. Chen, T.B., Lu, H.H.S., Lee, Y.S., Lan, H.J.: Segmentation of cDNA microarray images by kernel density estimation. Journal of Biomedical Informatics 41, 1021–1027 (2008)
7. Giannakeas, N., Fotiadis, D.I.: An automated method for gridding and clustering-based segmentation of cDNA microarray images. Comput. Med. Imaging Graph. 33, 40–49 (2009)
8. Bariamis, D., Maroulis, D., Iakovidis, D.: Unsupervised SVM-based gridding for DNA microarray images. Comput. Med. Imaging Graph. (2009)
9. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., et al.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286(5439), 531–537 (1999)
10. Sturn, A., Quackenbush, J., Trajanoski, Z.: Genesis: cluster analysis of microarray data. Bioinformatics 18(1), 207–208 (2002)
11. Chopra, P., Kang, J., Yang, J., Cho, H.J., Kim, H.S., Lee, M.G.: Microarray data mining using landmark gene-guided clustering. BMC Bioinformatics 9(92), 1–13 (2008)
12. Peterson, L.E., Coleman, M.A.: Logistic ensembles of Random Spherical Linear Oracles for microarray classification. Int. J. Data Min. Bioinform. 3(4), 297–382 (2009)
13. Axon GenePix Pro 7.1., http://www.moleculardevices.com
14. Eisen, M.: Scanalyze (1999), http://rana.lbl.gov/EisenSoftware.html
15. Ruosaari, S., Hollmen, J.: Image analysis for detecting faulty spots from microarray images. In: Lange, S., Satoh, K., Smith, C. (eds.) DS 2002. LNCS, vol. 2534, pp. 259–266. Springer, Heidelberg (2002)

16. Hautaniemi, S., Edgren, H., Vesanen, P., Wolf, M., Jrvinen, A.K., Yli Harja, O., et al.: A novel strategy for microarray quality control using bayesian networks. Bioinformatics 19(16), 2031–2038 (2003)
17. Bicego, M., Martínez, M.D.R., Murino, V.: A supervised data-driven approach for microarray spot quality classification. Pattern Anal. Applic. 8, 181–187 (2005)
18. Vapnik, V.: The nature of statistical learningn theory. Springer, Heidelberg (1995)
19. Liu, X., Zhang, L., Li, M., Zhang, H., Wang, D.: Boosting image classification with LDA-based feature combination for digital photograph management. Pattern Recognition 38(6), 887–901 (2005)
20. Baluja, S., Rowley, H.A.: Boosting sex identification performance. International Journal of Computer Vision 71(1), 111–119 (2007)
21. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, 2nd edn. Springer, Heidelberg (2009)
22. Valiant, L.G.: A theory of the learnable. Communications of the ACM 27, 1134–1142 (1984)
23. Schapire, R.E.: The Strength of Weak Learnability. Mach. Learn. 5(2), 197–227 (1990)
24. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55(1), 119–139 (1997)
25. Friedman, J.H., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. Annals of Statistics 28, 337–407 (2000)
26. Benediktsson, J.A., Swain, P.H.: Consensus theoretic classification methods. IEEE Trans. Systems, Man Cybernet. 22, 688–704 (1992)

A First Machine Learning Approach to Pronominal Anaphora Resolution in Basque

O. Arregi, K. Ceberio, A. Díaz de Illarraza,
I. Goenaga, B. Sierra, and A. Zelaia

University of the Basque Country
olatz.arregi@ehu.es

Abstract. In this paper we present the first machine learning approach to resolve the pronominal anaphora in Basque language. In this work we consider different classifiers in order to find the system that fits best to the characteristics of the language under examination. We do not restrict our study to the classifiers typically used for this task, we have considered others, such as Random Forest or VFI, in order to make a general comparison. We determine the feature vector obtained with our linguistic processing system and we analyze the contribution of different subsets of features, as well as the weight of each feature used in the task.

1 Introduction

Pronominal anaphora resolution is related to the task of identifying noun phrases that refer to the same entity mentioned in a document.

According to [7]: *anaphora, in discourse, is a device for making an abbreviated reference (containing fewer bits of disambiguating information, rather than being lexically or phonetically shorter) to some entity (or entities).*

Anaphora resolution is crucial in real-world natural language processing applications e.g. machine translation or information extraction. Although it has been a wide-open research field in the area since 1970, the work presented in this article is the first dealing with the subject for Basque, especially in the task of determining anaphoric relationship using a machine learning approach.

The first problem to carry out is the lack of a big annotated corpus in Basque. Mitkov in [12] highlights the importance of an annotated corpus for research purposes: *The annotation of corpora is an indispensable, albeit time-consuming, preliminary to anaphora resolution (and to most NLP tasks or applications), since the data they provide are critical to the development, optimization and evaluation of new approaches.*

Recently, an annotated corpus has been published in Basque with pronominal anaphora tags [2] and thanks to that, this work could be managed.

2 Related Work

Although the literature about anaphora resolution with machine learning approaches is very large, we will concentrate on those references directly linked to

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 234–243, 2010.

the work done here. In [20] they apply a noun phrase (NP) coreference system based on decision trees to MUC6 and MUC7 data sets ([15], [16]). It is usually used as a baseline in the coreference resolution literature.

Kernel functions to learn the resolution classifier are applied in [23]. They use structured syntactic knowledge to tackle pronoun resolution, and the results obtained for the ACE dataset show an improvement for all the different domains.

In [22] the authors propose kernel-based methods to resolve three coreference resolution subtasks (binding constraint detection, expletive identification and aliasing). They conclude that using kernel methods is a promising research direction to achieve state of the art coreference resolution results.

A rich syntactic and semantic processing is poposed in [5]. It outperforms all unsupervised systems and most supervised ones.

The state of the art of other languages varies considerably. In [18] they propose a rule-based system for anaphora resolution in Czech. They use the Treebank data, which contains more than 45,000 coreference links in almost 50,000 manually annotated Czech sentences. In [21] the author uses a system based on a loglinear statistical model to resolve noun phrase coreference in German texts. On the other hand, [13] and [14] present an approach to Persian pronoun resolution based on machine learning techniques. They developed a corpus with 2,006 labeled pronouns.

A similar work was carried out for Turkish [24]. They apply a decision tree and a rule-based algorithm to an annotated Turkish text.

3 Selection of Features

3.1 Main Characteristics of Pronominal Anaphora in Basque

Basque is not an Indo-European language and differs considerably in grammar from languages spoken in other regions around. It is an agglutinative language, in which grammatical relations between components within a clause are represented by suffixes. This is a distinguishing characteristic since morphological information of words is richer than in the surrounding languages. Given that Basque is a head final language at the syntactic level, the morphological information of the phrase (number, case, etc.), which is considered to be the head, is in the attached suffix. That is why morphosyntactic analysis is essential.

In this work we specifically focus on the pronominal anaphora; concretely, the demonstrative determiners when they behave as pronouns. In Basque there are not different forms for third person pronouns and demonstrative determiners are used as third person pronominals [11]. There are three degrees of demonstratives that are closely related to the distance of the referent: hau (this/he/she/it), hori (that/he/she/it), hura (that/he/she/it). As we will see in the example of Section 3.3 demostratives in Basque do not allow to infer whether the referent is a person (he, she) or it is an impersonal one (it).

Moreover, demostrative determiners do not have any gender in Basque. Hence, the gender is not a valid feature to detect the antecedent of a pronominal anaphora because there is no gender distinction in the Basque morphological system.

3.2 Determination of Feature Vectors

In order to use a machine learning method, a suitable annotated corpus is needed. We use part of the Eus3LB Corpus[1] which contains approximately 50.000 words from journalistic texts previously parsed. It contains 349 annotated pronominal anaphora.

In this work, we first focus on features obtainable with our linguistic processing system proposed in [1]. We can not use some of the common features used by most systems ([20], [17], [23]) due to linguistic differences. For example the gender, as we previously said. Nevertheless, we use some specific features that linguistic researchers consider important for this task.

The features used are grouped in three categories: features of the anaphoric pronoun, features of the antecedent candidate, and features that describe the relationship between both.

- Features of the anaphoric pronoun
  - f_1 - *dec_ana*: The declension case of the anaphor.
  - f_2 - *sf_ana*: The syntactic function of the anaphor.
  - f_3 - *phrase_ana*: Whether the anaphor has the phrase tag or not.
  - f_4 - *num_ana*: The number of the anaphor.
- Features of the antecedent candidate
  - f_5 - *word*: The word of the antecedent candidate.
  - f_6 - *lemma*: The lemma of the antecedent candidate.
  - f_7 - *cat_np*: The syntactic category of the NP.
  - f_8 - *dec_np*: The declension case of the NP.
  - f_9 - *num_np*: The number of the NP.
  - f_{10} - *degree*: The degree of the NP that contains a comparative.
  - f_{11} - *np*: Whether the noun phrase is a simple NP or a composed NP.
  - f_{12} - *sf_np*: The syntactic function of the NP.
  - f_{13} - *enti_np*: The type of entity (PER, LOC, ORG).
- Relational features
  - f_{14} - *dist*: The distance between the anaphor and the antecedent candidate. Its possible values are from 1 to 15, the maximum distance shown in the corpus from an anaphor to its antecedent. The distance is measured in terms of number of Noun Phrases.
  - f_{15} - *same_sent*: If the anaphor and the antecedent candidate are in the same sentence the value is 0, otherwise the value is 1.
  - f_{16} - *same_num*: Its possible values are 0, 1, 2, and 3. If the anaphor and the antecedent candidate agree in number the value is 3, otherwise the value is 0. When the number of the noun phrase is unknown the value is 1. If the noun phrase is an entity, its number is indefinite and the anaphor is singular, then the value is 2. This last case is needed in Basque because person entities do not have singular or plural tags, but indefinite tag.

[1] Eus3LB is part of the 3LB project [19].

In summary we would like to remark that we include morphosyntactic information in our pronoun features such as the syntactic function it accomplishes, the kind of phrase it is, and its number. We also include the pronoun declension case. We use the same features for the antecedent candidate and we add the syntactic category and the degree of the noun phrase that contains a comparative. We also include information about name entities indicating the type (person, location and organization). The word and lemma of the noun phrase are also taken into account. The set of relational features includes three features: the distance between the anaphor and the antecedent candidate, a Boolean feature that shows whether they are in the same sentence or not, and the number agreement between them.

3.3 Generation of Training Instances

The method we use to create training instances is similar to the one explained in [20]. Positive instances are created for each annotated anaphor and its antecedent. Negative instances are created by pairing each annotated anaphor with each of its preceding noun phrases that are between the anaphor and the antecedent. When the antecedent candidate is composed, we use the information of the last word of the noun phrase to create the features due to the fact that in Basque this word is the one that contains the morphosyntactic information.

In order to clarify the results of our system, we introduce the following example: **Ben Amor** *ere ez da Mundiala amaitu arte etorriko Irunera*, **honek** *ere Tunisiarekin parte hartuko baitu Mundialean.*

(**Ben Amor** *is not coming to Irun before the world championship is finished, since* **he** *will play with Tunisia in the World Championship*).

The word *honek* (he) in bold is the anaphor and *Ben Amor* its antecedent. The noun phrases between them are *Mundiala* and *Irunera*. The next table shows the generation of training instances from the sentence of the example.

Antecedent Candidate	Anaphor	Positive
Ben Amor	honek (he/it)	1
Mundiala	honek (he/it)	0
Irunera	honek (he/it)	0

Generating the training instances in that way, we obtained a corpus with 968 instances; 349 of them are positive, and the rest, 619, negatives.

4 Evaluation

In order to evaluate the performance of our system, we use the above mentioned corpus, with 349 positive and 619 negatives instances. Due to the size of the corpus, a 10 fold cross-validation is performed. It is worth to say that we are trying to increase the size of the corpus.

4.1 Learning Algorithms

We consider different machine learning paradigms from Weka toolkit [6] in order to find the best system for the task. The classifiers used are: SVM, Multilayer Perceptron, NB, k-NN, Random Forest (RF), NB-Tree and Voting Feature Intervals (VFI). We tried some other traditional methods like rules or simple decision trees, but they do not report good results for our corpus.

The SVM learner was evaluated by a polynomial kernel of degree 1. The k-NN classifier, $k = 1$, uses the Euclidean distance as distance function in order to find neigbours. Multilayer Perceptron is a neural network that uses backpropagation to learn the weights among the connections, whereas that NB is a simple probabilistic classifier based on applying Bayes' theorem, and NB-Tree generates a decision tree with naive Bayes classifiers at the leaves. Random forest and VFI are traditionally less used algorithms; however, they produce the best results for our corpus. Random forest is a combination of tree predictors, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [3]. VFI constructs feature intervals for each feature. An interval represents a set of values for a given feature, where the same subset of class values is observed. Two neighbouring intervals contain different sets of classes [4].

4.2 Overall Results

Table 1. shows the results obtained with these classifiers.

Table 1. Results of different algorithms

	Precision	Recall	F-measure
VFI	0.653	0.673	0.663
Perceptron	0.692	0.682	**0.687**
RF	0.666	0.702	0.683
SVM	0.803	0.539	0.645
NB-tree	0.771	0.559	0.648
NB	0.737	0.587	0.654
k-nn	0.652	0.616	0.633

The best result is obtained by using the Multilayer Perceptron algorithm, F-measure 68.7%.

In general, precision obtained is higher than recall. The best precision is obtained with SVM (80.3%), followed by NB-tree (77.1%). In both cases, the recall is similar, 53.9% and 55.9%.

These results are not directly comparable with those obtained for other languages such as English, but we think that they are a good baseline for Basque language. We must emphasize that only the pronominal anaphora is treated here, so actual comparisons are difficult.

5 Contribution of Features Used

Our next step is to determine the attributes to be used in the learning process. When there is a large number of attributes, even some relevant attributes may be redundant in the presence of others. Relevant attributes may contain useful information directly applicable to the given task by itself, or the information may be (partially) hidden among a subset of attributes [10].

To better understand which of the features used are more efficient, we evaluate the weight of attributes by different measurements: Information Gain, Relief algorithm, Symmetrical Uncertainty, Chi Squared statistic, and Gain Ratio. The order of features derive from each of the measurements is quite similar in all cases except for the Relief algorithm [8]. Although the first four features are the same in all cases (with slight order variations), the Relief algorithm shows a different order beyond the fifth feature, giving more weight to *word* or *lemma* features than to others relating to anaphor.

Fig. 1. shows the weight of these features taking into account all the measurements used.

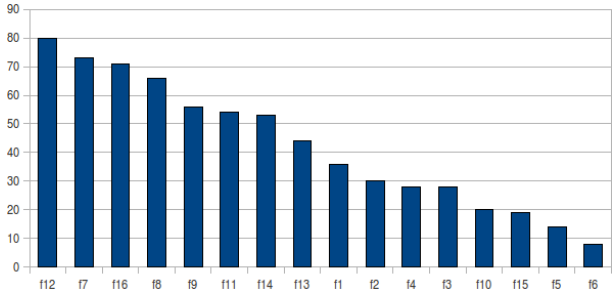


Fig. 1. The average weight of features

As expected, the features *word* and *lemma* do not contribute much to the classification process, and we can say that, in general, features relating to the anaphor are not very important for this task, while relational features like *same_num* (agreement in number) or *dist* (distance) appeared to be important. Moreover, all measurements show that features corresponding to the noun phrase are meaningful for this task, as indicated by other authors.

If we test the algorithms presented in Section 4.1, taking into account the new order of features, and considering smaller subsets of features, the results are similar to the originals. In general, decreasing the number of features gives lower results. The best result (70%) is obtained with 14 features: the original set without the features *word* and *lemma*.

Table 2. shows the best F-measure results obtained with the classifiers mentioned above, taking into account different feature subsets. Only five methods are shown here, due to the fact that results obtained with SVM and NB-tree are not meaningful. SVM method does not improve the first result (64.5%) and NB-tree provides similar results to the ones obtained by simple NB.

Table 2. Results of five algorithms with different number of features

Number of features	VFI	Perceptron	RF	NB	k-nn
16	0.663	0.678	0.683	0.654	0.633
15	0.669	0.669	0.678	0.656	0.648
14	0.671	**0.692** all - $\{f_1, f_2\}$	**0.7** all - $\{f_5, f_6\}$	0.665	0.655
13	0.670	0.678	0.679	0.663	0.666
12	0.669	0.671	0.677	0.665	0.662
11	0.672	0.670	0.690	0.666	**0.674**
10	0.675	0.679	0.674	**0.669**	0.656
9	0.674	0.687	0.679	0.666	0.665
8	0.674	0.672	0.682	0.661	0.661
7	0.677	0.668	0.661	0.655	0.644
6	**0.684**	0.652	0.664	0.650	0.640
5	0.673	0.645	0.652	0.640	0.625
4	0.655	0.619	0.628	0.632	0.600
3	0.646	0.639	0.661	0.619	0.616
2	0.629	0.635	0.626	0.607	0.617

Although the two best results were obtained with 14 features, 69.2% (perceptron) and 70% (RF), the set of attributes selected in both cases is different, since in the first case the best selection of features is produced by the relief algorithm (all features except *sf_ana* and *dec_ana*), and in the second case features were chosen following the order established by the Gain Ratio measurement (all features except *word* and *lemma*). For the rest of the algorithms, the best results are obtained by using a smaller set of attributes (from 6 to 11); nevertheless these results are lower than those mentioned above. For all the algorithms we obtained a higher value than the original F-measure. Table 3. shows these values.

Table 3. Results obtained with different subsets of features

	original F-measure	best F-measure	Number of features
VFI	0.663	0.684	6
Perceptron	0.687	0.692	14
RF	0.683	**0.700**	14
NB	0.654	0.669	10
k-nn	0.633	0.670	11

For the k-NN method the measurement which offers the best results is, in most cases, the Relief algorithm. This result was expected as this algorithm evaluates the weight of a feature by repeatedly sampling an instance and considering the value of the given feature for the nearest instance of the same and different class. So, given an instance, Relief algorithm searches for its two nearest neighbours,

and the k-NN algorithm is based on the same idea. The selection of the nearest neighbours is crucial in Relief. The purpose is to find the nearest neighbours with respect to important attributes [9].

5.1 The Contribution of Single Attributes

If we use a single attribute each time for the classification process, we can determine that the best attribute is *sf_np*, that is, the syntactic function of the noun phrase, with an F-measure equal to 0.480 but a precision of 0.905.

Table 4. shows the results obtained for this test applying Random Forest algorithm. Unsurprisingly many of the attributes result in zero. It should be noted that as in other works [20], selected attributes provide high values for precision, although the recall is very low. The first four attributes of the table, which are the same as those selected by the measurements introduced at the beginning of this section, provide a precision above 65%, reaching to 90% in the case of the first attribute (*sf_np*). In contrast, the F-measure values are lower than 50%.

Table 4. Results obtained using just one attribute at a time

	Precision	Recall	F-measure
sf_np	0.905	0.327	0.480
cat_np	0.659	0.309	0.421
same_num	0.811	0.123	0.214
dec_np	0.837	0.249	0.384
lemma	0.421	0.381	0.400
word	0.378	0.347	0.362
dist	0.364	0.011	0.022
Rest of attributes	0.000	0.000	0.000

6 Conclusions and Future Work

This is the first study carried out on resolution of pronominal anaphora in Basque using a machine learning approach. It has been a useful start in defining criteria for anaphora resolution. The results obtained from this work will be helpful for the development of a better anaphora resolution tool for Basque.

We consider seven machine learning algorithms for our first approach in order to decide which kind of method can be the best for this task. The best results are obtained with two classifiers (Random Forest and VFI) which are not the most used for this task in other languages. This may be due to the chosen feature set, the noise of the corpus, and the Basque language characteristics. Traditional methods like SVM, give us a good precision but an F-measure four points below the best system. Anyway, the corpus used in this work is quite small, so we think that the results we obtain can be improved with a larger corpus.

We also analyzed the contribution of features used in order to decide which of them are important and which are not. With a good combination of features we obtain an F-measure of 70%, which is the best result obtained in this work.

There are several interesting directions for further research and development based on this work. The introduction of other knowledge sources to generate new features and the use of composite features can be a way to improve the system.

The combination of classifiers has been intensively studied with the aim of improving the accuracy of individual components. We intend to apply a multiclassifier based approach to this task and combine the predictions generated applying a Bayesian voting scheme.

We plan to expand our approach to other types of anaphoric relations with the aim of generating a system to determine the coreference chains for a document.

Finally, the interest of a modular tool to develop coreference applications is unquestionable. Every day more people research in the area of the NLP for Basque and a tool of this kind can be very helpful.

Acknowledgments

This work was supported in part by KNOW2 (TIN2009-14715-C04-01) and Berbatek (IE09-262) projects.

References

1. Aduriz, I., Aranzabe, M.J., Arriola, J.M., Daz de Ilarraza, A., Gojenola, K., Oronoz, M., Uria, L.: A Cascaded Syntactic Analyser for Basque. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 124–134. Springer, Heidelberg (2004)
2. Aduriz, I., Aranzabe, M.J., Arriola, J.M., Atutxa, A., Daz de Ilarraza, A., Ezeiza, N., Gojenola, K., Oronoz, M., Soroa, A., Urizar, R.: Methodology and steps towards the construction of EPEC, a corpus of written Basque tagged at morphological and syntactic levels for the automatic processing. In: Wilson, A., Archer, D., Rayson, P. (eds.) Language and Computers, Corpus Linguistics Around the World, Rodopi, Netherlands, pp. 1–15 (2006)
3. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
4. Demiroz, G., Guvenir, A.: Classification by voting feature intervals. In: 9th European Conference on Machine Learning, pp. 85–92 (1997)
5. Haghighi, A., Klein, D.: Simple Coreference Resolution with Rich Syntactic and Semantic Features. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 1152–1161 (2009)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
7. Hirst, G.: Anaphora in Natural Language Understanding. Springer, Berlin (1981)
8. Kira, K., Rendell, L.A.: A Practical Approach to Feature Selection. In: Ninth International Workshop on Machine Learning, pp. 249–256 (1992)
9. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: European Conference on Machine Learning, pp. 171–182 (1994)
10. Kononenko, I., Hong, S.J.: Attribute Selection for Modeling. Future Generation Computer Systems 13, 181–195 (1997)

11. Laka, I.: A Brief Grammar of Euskara, the Basque Language. Euskarako errektoreordetza, EHU (2000), http://www.ehu.es/grammar
12. Mitkov, R.: Anaphora resolution. Longman, London (2002)
13. Moosavi, N.S., Ghassem-Sani, G.: Using Machine Learning Approaches for Persian Pronoun Resolution. In: Workshop on Corpus-Based Approaches to Coreference Resolution in Romance Languages. CBA 2008 (2008)
14. Moosavi, N.S., Ghassem-Sani, G.: A Ranking Approach to Persian Pronoun Resolution. Advances in Computational Linguistics. Research in Computing Science 41, 169–180 (2009)
15. MUC-6.: Proceedings of the Sixth Message Understanding Conference (MUC-6). Morgan Kaufmann, San Francisco, CA (1995)
16. MUC-7.: Proceedings of the Seventh Message Understanding Conference (MUC-7). Morgan Kaufmann, San Francisco, CA (1998)
17. Ng, V., Cardie, C.: Improving Machine Learning Approach to Coreference Resolution. In: Proceedings of the ACL, pp. 104–111 (2002)
18. Nguy, Zabokrtský: Rule-based Approach to Pronominal Anaphora Resolution Method Using the Prague Dependency Treebank 2.0 Data. In: Proceedings of DAARC 2007 (6th Discourse Anaphora and Anaphor Resolution Colloquium) (2007)
19. Palomar, M., Civit, M., Díaz, A., Moreno, L., Bisbal, E., Aranzabe, M.J., Ageno, A., Mart, M.A., Navarro, B.: 3LB: Construcción de una base de datos de árboles sintáctico-semánticos para el catalán, euskera y español. XX. Congreso SEPLN, Barcelona (2004)
20. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A Machine Learning Approach to Coreference Resolution of Noun Phrases. Computational Linguistics 27(4), 521–544 (2001)
21. Versley, Y.: A Constraint-based Approach to Noum Phrase Coreference Resolution in German Newspaper Text. In: Konferenz zur Verarbeitung Natrlicher Sprache KONVENS (2006)
22. Versley, Y., Moschitti, A., Poesio, M., Yang, X.: Coreference System based on Kernels Methods. In: Proceedings of the 22nd International Coreference on Computational Linguistics (Coling 2008), Manchester, pp. 961–968 (2008)
23. Yang, X., Su, J., Tan, C.L.: Kernel-Based Pronoun Resolution with Structured Syntactic Knowledge. In: Proc. COLING/ACL 2006, Sydney, pp. 41–48 (2006)
24. Yldrm, S., Klaslan, Y., Yldz, T.: Pronoun Resolution in Turkish Using Decision Tree and Rule-Based Learning Algorithms. In: Human Language Technology. Challenges of the Information Society. LNCS. Springer, Heidelberg (2009)

A Holonic Multi-agent Model for Oil Industry Supply Chain Management

Fernando J.M. Marcellino and Jaime S. Sichman

LTI, University of São Paulo
Av. Prof. Luciano Gualberto 158 tv.3
05508-970 São Paulo SP, Brazil
{fmarcellino,jaime.sichman}@poli.usp.br

Abstract. Very few industries can benefit more from maximizing supply chain efficiencies than the oil and gas companies [1]. However, the behavior of such systems is too complex to be treated analytically. On the other hand, multi-agent systems show great similarity with respect to supply chains, and provide the flexibility to model the complexities and dynamics of real world chains without excessive simplifying assumptions. Since supply chain management deals with a recursive structure, it becomes appropriate to employ holonic agents for its modeling. Thus, this work aims at developing a holonic multi-agent model for a typical oil industry supply chain. A simple case study is analyzed as a proof-of-concept of the proposed approach. The obtained results motivate us to continue extending the model in order to make it more realistic.

Keywords: Oil Industry Supply Chains; Multi-Agent Systems; Holonic Agents.

1 Introduction

Supply chain management (SCM) can be defined as the configuration, coordination and continuous improvement of an organized set of operations. Its goal is to provide maximum customer service at the lowest cost possible, where a customer is anyone who uses the output of a process. Since the goal of a company is to maximize profits, it must weigh the benefits versus the costs of its decisions along the supply chain. Very few industries can benefit more from maximizing supply chain efficiencies than the oil and gas companies [1]. In this industry, there is a need to ensure that each entity along the supply chain can respond quickly to the exact needs of its customers. On the other hand, one of its weaknesses is that each entity is likely to act in its best interests to optimize its own profit. In general, that doesn't meet the goal of the entire supply chain. Integrating the management decisions of different parts of the supply chain may provide huge gains and benefits, which are usually concealed by the pitfalls of short-term and local angles. Historically, the decisions of business supply chains have been highly concentrated, and researchers have used analytical approaches focusing on how to improve the efficiency of the individual entities along the chain instead of improving its combined performance. Although great

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 244–253, 2010.

corporations own, in general, a global planning application that tries to take into account the main activities and goals of its components, the complexity of the resulting system makes an integration of these latter an unfeasible task. In addition, even if a centralized planning system is simple enough to work, the frequent and unforeseeable changes in the business environment render its results obsolete and useless very fast. Since multi-agent systems show great similarity with supply chains, multi-agent systems become a suitable means to manage the supply chain with a better performance than the one attained by the current human-software approach. Furthermore, the recursive nature of supply chains suggests the use of holonic agents, which are fractal structures that deal better with the communication between different abstraction levels of the chain.

The objective of this paper is to propose a preliminary framework of a holonic multi-agent model for the oil industry supply chain management. The next section synthesizes the basic concepts involved in the work. Section 3 describes the proposed model. Then, the following section is dedicated to a case study, where the proposed approach is compared with a usual way of managing oil supply chains. Finally, the paper concludes with a summary of the current results, and an outlook on future research activities.

2 Basic Concepts

A typical *oil industry supply chain* includes exploration of new petroleum (crude oil) reservoirs, drilling of crude wells, crude extraction at onshore and offshore platforms, its transport to the refineries, the refining of the crude oil (raw material) in the refineries in order to produce the final products (petroleum derivatives, such as gasoline and diesel), the transport of those final products to distribution terminals, where they are dispatched to distribution companies, and finally the delivery of the derivatives to the final customers (e.g. gas stations). There are different types of crude oil, with distinct qualities. Each of them has a specific production profile, yielding definite proportions of each derivative product. As to the transport of crude oil and its derivatives, it is carried out by ships, trains, trucks, and mainly by pipelines.

An important component in supply chain analysis is the choice of performance measures, which are used to determine the efficiency of a system, or to compare alternative ones. The literature categorizes these measures as either qualitative or quantitative. In general, quantitative measures are related to monetary values, as cost and profit, whereas qualitative ones are based on customer satisfaction [2]. Traditionally, mathematical optimization models are used to solve supply chain management problems; however, supply chains are too complex to allow realistic models to be evaluated analytically.

Multi-Agent Systems (MAS) offer such useful features as parallelism, robustness and scalability. They are highly applicable in domains where centralized approaches meet their limits [3]. They are composed of autonomous, reactive, proactive, and interacting entities called *agents*, engaged in the realization of a joint goal. Negotiation is a fundamental capability of agents to reach such a goal. The agent-based framework may lead to more effective integration of production, logistics, and

trading processes. Since a supply chain is a network of facilities, each agent can be assigned to model a facility, and relationships can be defined as links to connect these agents. Some works have already proposed the use of MAS in supply chains, like [4]. Compared to analytical techniques, agent-based simulation provides the flexibility to model the complexities and dynamics of real world supply chains without having to perform excessive simplifying assumptions [5].

Although MAS has become a natural tool for modeling and simulating complex systems, these systems usually contain a great number of entities interacting among themselves, and acting at different levels of abstraction. In this context, it seems unlikely that MAS will be able to faithfully represent complex systems without multiple granularities. That's why *holonic systems* have attracted the attention of researchers [6]. The term *holon* was coined by Arthur Koestler [7], based on the Greek word *holos* for whole and *on* for part. Thus, a holon is a self-similar or fractal structure that consists of several holons as components, and is itself a part of a greater whole. Gerber et al. [8] propose three types of structures for holons, which vary with respect to the autonomy of the members. The *moderated group* is the intermediary structure, which was chosen for this work due to its greater flexibility. It specifies a holonic organization with two main *roles*: *head* and *part*. The *head* is a kind of moderator of the holon, represents its shared intentions and negotiates them with agents outside the holon, as well as with those inside it. Only the *head* can communicate with the outer world. On the other hand, the *part* is any other member of the holon. Figure 4 outlines the holonic organization of this work.

3 Proposed Model

This work focuses on an oil industry supply chain, which starts at the crude oil extraction, and finishes when the crude derivative products are delivered to distribution companies, which are considered here as final customers. Thus, all activities that occur before, such as exploration and drilling, as well as those that occur after, such as the delivery of products to the gas stations, are out of the scope of this work. The supply of crude and its derivatives must be made preferentially by the oil company itself, which may be a single verticalized petroleum enterprise or a set of cooperating companies of the oil business. Henceforth, in this work we will call *extended enterprise* (EE) the general situation that comprises both cases. If necessary or eligible, an EE can purchase from the *spot market* (SM), which is able to provide any extra demands of crude oil and its derivatives at higher prices. In the same way, the SM can buy any exceeding inventories of those items at lower prices. An EE operation area is spread geographically, and this physical space is visualized as a partition of regions, which are in their turn grouped into continents, and these latter are finally gathered into a global area. A region is divided into trading areas or oil extraction areas. A refinery or distribution terminal is responsible for each of the first areas, which serve specific final customers (distributing companies). On the other hand, each of the latter areas is composed of oil extraction platforms, that yield a certain type of crude oil at a rate; this rate is considered constant for the time scale of this work. All the areas are connected by transportation modals, like ships and

pipelines. Furthermore, each area owns a specific logistic entity, which is responsible for the transportation planning of crude oil and derivatives, which, henceforward, will be named just products. Every transportation modal has a set of routes, which connect two entities of the chain (platforms, refineries, terminals and SM) and has a maximum transport capacity. If the connected entities belong to a same region, then the transport is managed by its regional logistics. Otherwise, it is managed by the continental or even the global logistics, depending on the range of the particular route. Since the facilities are connected through a transportation network, they can cooperate with each other to supply the different trading areas, instead of being restricted to their own ones. A refinery can produce multiple derivatives, and it does so according to different production plans, which are characterized by processing a definite quantity of a particular type of crude oil and producing a certain quantity of each resulting derivative. Figure 1 illustrates an example of the described supply chain.

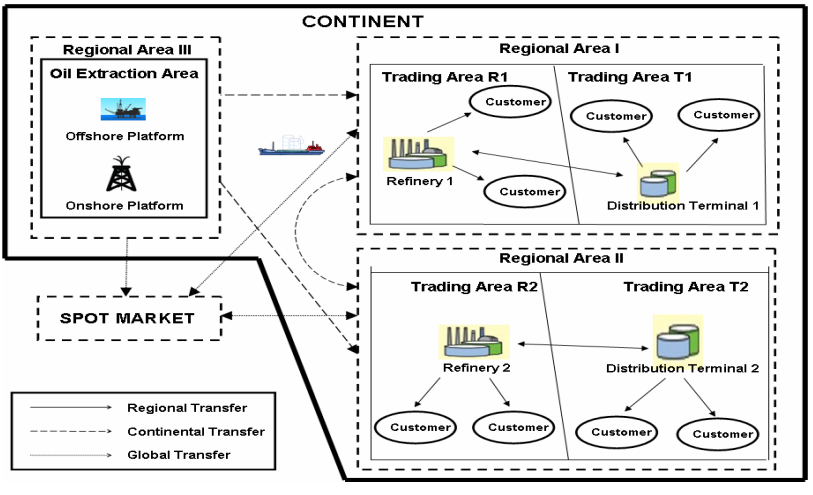


Fig. 1. Example of an oil supply chain

A general model should propose a multi-period time approach, where the supply chain system evolves over a given time horizon H, which is divided into several periods. Partially based on past decisions, current decisions are made at the beginning of each period. The sequence of such decisions along H will be called a *decision strategy* $\mathcal{S}$, where decision means the assignment of values to all the *decision variables* of the model. However, as an initial model, this work proposes a single period time approach, in order to get rid of the time modeling difficulties, and also to restrict the number of *decision variables*. For a further decrease of the model complexity, the inventory management is not treated in this work either. Thus, the remaining *decision variables* are only the *production plans* adopted by each refinery, which are assigned to values from discrete sets generated by the corresponding production planner of each refinery. The supply chain manager's motivation is to choose a *decision strategy* $\mathcal{S}$ such that the performance measure could be as good as

possible. In this work, the chosen performance measure is the total profit of the whole supply chain during H, which can be written as:

$$Profit(\mathcal{S}) = Income(\mathcal{S}) - Cost(\mathcal{S}) \quad (1)$$

$$Income(\mathcal{S}) = CrudeIncome(\mathcal{S}) + DerivativeIncome(\mathcal{S}) \quad (2)$$

$$Cost(\mathcal{S}) = CrudeCost(\mathcal{S}) + Freight(\mathcal{S}) + Penalty(\mathcal{S}) \quad (3)$$

In these formulae, *CrudeIncome* is the income obtained by selling the surplus petroleum to SM, *DerivativeIncome* comes from the sale of all the derivatives, *CrudeCost* is the cost resulting from crude extraction or its purchase from SM, *Freight* is the total freight with the transportation of crude and derivatives, and *Penalty* is the loss due to the purchase of more expensive products from SM to supply the demands of customers, who pay the regular price. As can be inferred from the previous text, the inventory costs are not considered in this initial model. Moreover, the production costs of derivatives in the refineries are not taken into account either.

One of the basic assumptions of the model is that all the customer demands must be supplied. That is always possible due to the flexibility offered by SM, which can buy or sell unbounded quantities of any product. However, it generates a cost, which is represented by the term *Penalty* of equation 3. Therefore, the objective is to maximize the total profit with the customer satisfaction automatically guaranteed.

In this work, we propose a holonic multi-agent system to model the supply chain. It is divided into 2 submodels: *organization model* and *interaction model*, which will be described respectively in the next subsections.

3.1 Organization Model

As a result of the features of a holonic system, the organization model can be represented in two dimensions: a *holonic organization*, which is common to all the holonic systems, and an *internal organization*, which is specific to a problem domain [6]. Thus, a same agent can have a role in the holonic organization, and another role in the internal organization. The holonic organization is composed basically of the roles *head* and *part*, whereas, in general, the internal organization comprises more roles, which are associated with physical entities, services and functions of the supply chain. In addition to these specific roles of the internal organization, two generic roles are shared by several of them. The first is the *supplier* role, which refers to any entity that can provide products to other entity of the chain. The other is the *client* role, which is played by any entity which needs these products. Figure 2 presents the UML class diagram, which represents the roles of the organization and the relationships between them. The refinery is modeled as a terminal with a *derivative producer*, and the *logistics* is the *head* of the respective holon in all levels of the chain, i.e., the *internal logistics* for the holon refinery or terminal, the *regional logistics* for the holon *region*, and so on, up to the *global logistics* for the holon *global*. Each *logistics is* responsible for the balance of products between the *suppliers* and the *clients*. The *external logistics*, which represents all the *logistics* roles, except the internal one, is responsible for the transport planning as well. As to the *production planner* role, it is played inside the refinery either by a software or a team of human experts, which

support the *derivative producer* role. The *sales* role, in turn, is responsible for the forecast of the total demand for each derivative at the corresponding trading area.

Although the holonic paradigm allows a smaller granularity, with agents representing operational units of the refinery, in this work we will limit the detail level to the *derivative producer*, which puts up a discret set of production plans obtained from the *production planner*.

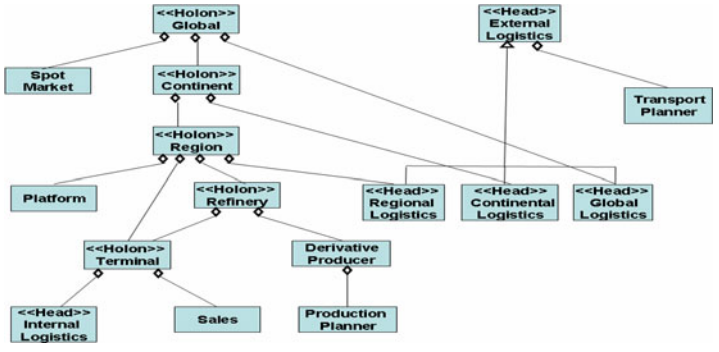


Fig. 2. Class diagram of the organization model

3.2 Interaction Model

The flexibility and autonomy provided by the holonic organization demand an interaction protocol to guide the negotiation between several agents, which should lead to the choice of a production plan for each refinery. This protocol is coordinated by each *head* agent, according both to the recursive nature of the holonic architecture, and to the details of the oil supply chain in focus. The model proposes a single protocol to be used in all the recursion levels of the chain. It is illustrated by the UML collaboration diagram in Figure 3, where the agents may play the generic roles *supplier* or *client*, which are mutually exclusive, and the *logistics* role, which has also the *head* role in all the recursive levels of the organization. In an abridged way, the protocol starts when the *logistics* agent receives messages from all the *client* agents informing their needs of products, and also from all the *supplier* agents informing their availability of those items[1], i.e., the surplus products they wish to drain. Based on the knowledge that the freight for transport between holons of the upper level is more expensive than the one for transport inside the holon, the *logistics* agent tries to match the needs to the availabilities for each product, aiming at the maximum balance between internal *suppliers* and *clients*, in order to maximize the total profit. In the case of *external logistics*, it does so taking into account the transport information. Then, the *logistics* agent verifies the pending needs or availability of each product, and if it exists, a message is sent to the *head* of the superholon (higher level *logistics* agent), informing such a need or availability. Inside a refinery, the *supplier* role is

[1] Although the needs or availability of each product are continuous quantities in the real world, they are modeled as discrete entities here. This choice, however, doesn't mean a less reliable representation of the reality, since the quality of the model will hinge on the adoption of an appropriate granularity.

represented by the *derivative producer*, whose *production planner* proposes several alternative *production plans*. Thus, the *internal logistics* repeats the same activities for each possible *production plan* and informs its superholon accordingly. The same behavior is repeated recursively from one layer to its higher neighbor, up to the global level, where all the possible combinations are compared by the *global logistics*, in order to choose the one associated with the maximum total profit. Figure 4 depicts the recursive levels of the model.

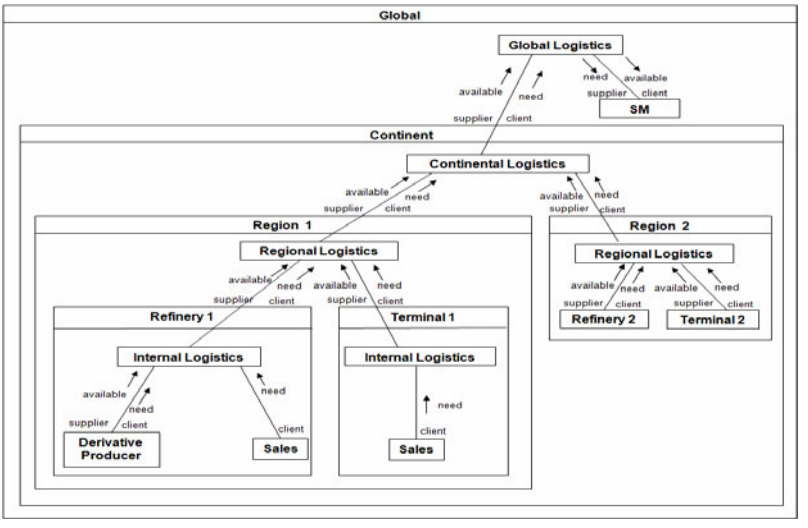


Fig. 3. Collaboration diagram of the holonic interaction model for each product

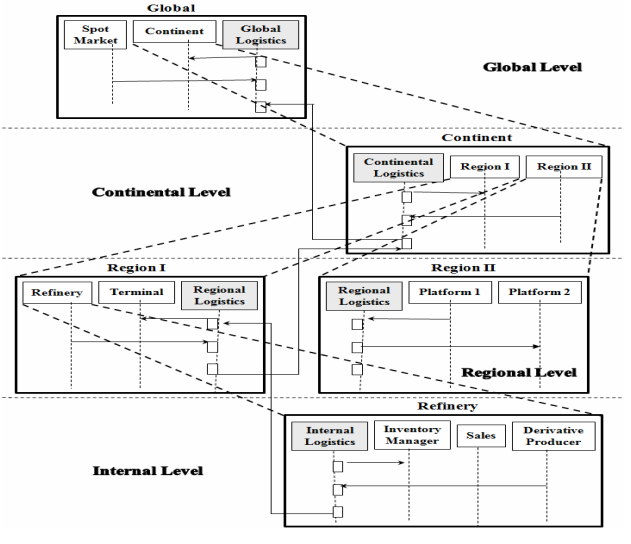


Fig. 4. Holonic organization with the *head* holons highlighted

4 Case Study

It was analyzed a case study, which was simple but representative, since it includes all the relevant entities of the chain, plus a significant set of products. It contains one continent with three regions, and one overseas SM. The first two regions comprise one refinery and one terminal, whereas the third region contains only one oil extraction area. Inside the regions, entities are connected by pipelines, but regions and SM are connected to each other by ships. The maximum transport capacity of each possible route is unlimited for the sake of simplicity. The refineries produce three derivatives (gasoline, diesel and naphtha) by processing three types of crude oil (pet1, pet2 and pet3). The first two types are extracted only by the EE, whereas the third is obtained only from SM. This latter can buy the crude oil types pet1 and pet2, but can't sell them. Furthermore, SM can supply any derivative, and also purchase them. The refineries can operate according to three production plans, which are specific to each refinery: plan A, plan B, and plan C. As to the transport, the freight fee is 1 for transfers inside a region, 2 inside a continent, and 10 for global transfers involving SM. Table 1 shows the other scenario parameters. It is possible to realize by the product prices that it isn't worth buying from or selling to the SM, unless it is strictly necessary in order to supply the customer demands. Figure 1 illustrates exactly the supply chain of the case study. Although it is not a real situation, it is representative and fits for a proof-of-concept, which is accomplished by comparing the proposed model with a usual approach to manage the oil supply chain, which will be called *local optimization*. The results of applying both approaches are presented next.

Table 1. Petroleum (left) and its derivatives (right) parameters

Petroleum types		Pet1	Pet2	Pet3
Extraction costs		10	15	-
Extraction volumes		30	40	-
SM prices:	buy	30	35	-
	sell	-	-	80

Products		Gaso	Diesel	Napht
Local Demand	Refinery 1	2	1	0
	Refinery 2	4	3	1
	Terminal 1	3	2	3
	Terminal 2	5	4	5
Prices	Continental	200	160	320
	SM buy	150	120	240
	SM sell	250	200	400

4.1 Local Optimization Approach

Most of the actual supply chains apply this approach, where managers have a selfish and limited angle, without taking into account the best for the EE as a whole. That limitation results from the complexity of the supply chain and the difficulty of dealing with the swift changes in its environment. Thus, the choice of the production plan in each refinery is based on local data and simplified assumptions, as it is shown in Table 2. It describes the three production plans of both refineries, one per processed crude oil type. Referring to equations 2 and 3, column *Cost* represents *CrudeCost* + petroleum *Freight*, and *Income* is *DerivativeIncome*, assuming the selling of all products to EE market (without SM) is the best scenario. Since from a local perspective it isn't known whether the products will be sold to the same region area or to another one, product freights can't be forecasted. Thus, the plan B is chosen in both refineries, since it leads to the highest supposed profit. Next, the logistics has to plan

the distribution of the available refined derivatives in order to supply all customer demands. In case of product unbalances, it will trade with other entities. Thus, the final recommended decision strategy points out a total profit of 5291 (1716 + 3575).

Table 2. Decision strategies for the production plans using the local optimization approach

		Input Crude			Output Product			Income	Cost	Profit
		Petl	Pet2	Pet3	Gaso	Diesel	Napht			
	Plan A	10			3	2	1	1240	120	1120
Ref. 1	**Plan B**		**12**		**4**	**3**	**2**	**1920**	**204**	**1716**
	Plan C			15	6	5	3	2960	1350	1610
	Plan A	20			6	5	3	2960	240	2720
Ref. 2	**Plan B**		**25**		**8**	**7**	**4**	**4000**	**425**	**3575**
	Plan C			30	12	10	6	5920	2700	3220

4.2 Holonic Multi-Agent Approach

We have carried out some experiments using the proposed holonic multi-agent model, which provides a wide negotiation between the supply chain entities in several levels, aiming at the best for the whole EE. Each production plan of a refinery is taken into account, and each *logistics* agent tries to balance its availability and its needs. Table 3 shows all the possible decision strategies (DS), which are defined by the chosen production plans of both refineries (R1, R2). For each product it is shown: demands (De) of the trading areas of refineries and terminals (T1, T2), productions (Pr) of the refineries, volumes transferred between chain entities (X>Y), *DerivativeIncome* (DI), *Freight* cost for products (Fr), and *Penalty* (Pen) due to the purchase of products from SM. In addition, for each type of crude oil it is shown the *CrudeIncome* (CI) of the crude oil sold to SM, the *CrudeCost* (CC) of the crude oil extracted by the EE or bought from SM, and the *Freight* cost for crude oils (Fr). As a result of negotiation, the model leads to the conclusion that the best decision strategy is to adopt production plan C in both refineries (CC), since it yields the highest profit (5918). This result disagrees with the conclusion of the local optimization approach, which recommends plan B in both refineries. By the way, it is possible to verify in Table 3 that the real profit of strategy BB is 5559, rather than 5291 as predicted by the first approach. This discrepancy comes from the myopia of that approach, which is concerned neither about the sales of surplus crude oil by the production area, nor about the penalties incurred by not having product enough to supply all customer demands.

5 Conclusion and Future Work

This work proposes a holonic multi-agent model for the management of a typical oil industry supply chain. This model consists of an organization model, based on the holonic organization of the kind *moderated group*, and an interaction model, which reproduces the recursive nature of that sort of organization. As a first validation of the model, a representative case study was analyzed. Despite its simplicity, it fulfilled its purpose as a proof-of-concept of the proposed model. The results motivate us to continue extending the model in order to make it more realistic. Thus, in the next step the model will include a multi-period time approach and the inventory management

by adding new decision variables associated with the choice of the inventory policy to be adopted in each refinery and terminal. In addition, the transport constraints should be properly treated in the model. The inclusion of such constraints, as well as the consideration of limited availabilities or needs of SM, will restrict the space of possibilities to be explored aiming at the maximum total profit. Moreover, the nature of the problem suggests the use of a constraint network to represent the interaction model. Another aspect to be investigated is how the model scales up with respect to the number of agents, and its feasibility to be connected in real time with supply chain legacy systems.

Table 3. Decision strategies for the production plans using the holonic multi-agent approach

DS	Prd	R1		T1	R2		T2	Transfer								Prods			Crude			Profit
		De	Pr	De	Pr	Pr	De	R1>T1	R2>T2	R1>T2	R2>T1	R1>SM	R2>SM	SM>T1	SM>T2	Dl	Fr	Pen	Cl	Cos	Fr	
AA	G	2	3	3	4	6	5	1	2	0	0	0	0	2	3	1800	53	250	-	-	-	3931
	D	1	2	2	3	5	4	1	2	0	0	0	0	1	2	1120	33	120	1000	300	60	
	N	0	1	3	1	3	5	1	2	0	0	0	0	2	3	1280	53	400	-	-	-	
AB	G	2	3	3	4	6	5	1	4	0	0	0	0	2	1	2200	35	150	-	-	-	4866
	D	1	2	2	3	5	4	1	4	0	0	0	0	1	0	1440	15	40	775	475	70	
	N	0	1	3	1	3	5	1	3	0	0	0	0	2	2	1600	44	320	-	-	-	
AC	G	2	3	3	4	12	5	1	5	0	0	0	1	0	0	2950	20	0	-	-	-	5377
	D	1	2	2	3	10	4	1	4	0	0	0	2	0	0	1840	27	0	1400	2500	320	
	N	0	1	3	1	6	5	1	5	0	0	0	0	2	0	2240	26	160	-	-	-	
BA	G	2	4	3	4	6	5	2	2	0	0	0	0	1	3	2000	44	200	-	-	-	4624
	D	1	3	2	3	5	4	2	2	0	0	0	0	0	2	1280	24	80	900	380	64	
	N	0	2	3	1	3	5	2	2	0	0	0	0	1	3	1600	44	320	-	-	-	
BB	G	2	4	3	4	8	5	2	4	0	0	0	0	1	1	2400	26	100	-	-	-	5559
	D	1	3	2	3	7	4	2	4	0	0	0	0	0	0	1600	6	0	675	555	74	
	N	0	2	3	1	4	5	2	3	0	0	0	0	1	2	1920	35	240	-	-	-	
BC	G	2	4	3	4	12	5	2	5	0	0	0	2	0	0	3100	29	0	-	-	-	5854
	D	1	3	2	3	10	4	2	4	0	0	0	3	0	0	1960	36	0	1300	2580	324	
	N	0	2	3	1	6	5	2	5	0	0	0	0	1	0	2560	17	80	-	-	-	
CA	G	2	6	3	4	6	5	3	2	1	0	0	0	0	2	2400	27	100	-	-	-	5120
	D	1	5	2	3	5	4	2	2	2	0	0	0	0	0	1600	8	0	1200	1400	190	
	N	0	3	3	1	3	5	3	2	0	0	0	0	0	3	1920	35	240	-	-	-	
CB	G	2	6	3	4	8	5	3	4	1	0	0	0	0	0	2800	9	0	-	-	-	5859
	D	1	5	2	3	7	4	2	4	0	0	2	0	0	0	1840	26	0	975	1575	200	
	N	0	3	3	1	4	5	3	3	0	0	0	0	0	2	2240	26	160	-	-	-	
CC	G	2	6	3	4	12	5	3	5	0	0	1	3	0	0	3400	48	0	-	-	-	5918
	D	1	5	2	3	10	4	2	4	0	0	2	3	0	0	2200	56	0	1600	3600	450	
	N	0	3	3	1	6	5	3	5	0	0	0	0	0	0	2880	8	0	-	-	-	

References

1. Chima, M.C.: Supply-Chain Management Issues in the Oil and Gas Industry. Journal of Business & Economics Research 5(6) (June 2007)
2. Beamon, B.M.: Supply Chain Design and Analysis: Models and Methods. International Journal of Production Economics 55, 281–294 (1998)
3. Graudina, V., Grundspenkis, J.: Technologies and Multi-Agent System Architectures for Transportation and Logistics Support: An Overview. In: International Conference on Computer systems and Technologies – CompSysTech, pp. IIIA.6-1–IIIA.6-2 (2005)
4. Swaminathan, J.M., Smith, S.F., Sadeh, N.M.: Modeling Supply Chain Dynamics: a Multiagent Approach. Decision Sciences 29(3) (Summer 1998)
5. Mele, F.D., Guilln, G., Espua, A., Puigjaner, L.: A Simulation-Based Optimization Framework for Parameter Optimization of Supply-Chain Networks. Industrial & Engineering Chemistry Research 45(9) (2006)
6. Hilaire, V., Koukam, A., Rodriguez, S.: An Adaptative Agent Architecture for Holonic Multi-Agent Systems. ACM Transactions on Autonomous and Adaptive Systems 3(1), Article 2 (March 2008)
7. Koestler, A.: The Ghost in the Machine. Hutchinson & Co., London (1967)
8. Gerber, C., Siekmann, J., Vierke, G.: Holonic Multi-Agent Systems. In: Research Report RR-99-03, DFKI (1999)

Moral Minds as Multiple-Layer Organizations

Helder Coelho[1], António Carlos da Rocha Costa[2], and Paulo Trigo[3]

[1] LabMAg, Faculdade de Ciências da Universidade de Lisboa, 1749-016 Lisboa, Portugal
hcoelho@di.fc.ul.pt
[2] C3/PPGMC, Universidade Federal do Rio Grande, Rio Grande, RS, Brazil
ac.rocha.costa@gmail.com
[3] LabMAg, Instituto Superior de Eng. de Lisboa, DEETC, 1959 Lisboa, Portugal
ptrigo@deetc.isel.ipl.pt

Abstract. The design of a moral decision apparatus is always associated with the appropriate definition of the set of qualities and virtues of an agent which emerge from their feelings and desires. But, there are much more elements along the process of moral judgement and there is no single linear sequence of effects. Therefore, a moral mind is a n-layer organization where the decision apparatus plays a central role in fixing the policy of choice. The idea of the agent organization will be illustrated in a devious negotiation scenario.

Keywords: moral architecture, values and norms, behaviour regulation, morality reconsideration, POMDP modelling.

1 Introduction

The way an agent determines what is right and wrong illuminates the role character plays. Such procedure is not only concerned with the relation between passion and deliberation, but also with how the agent sizes up practical social situations and computes what it is best to do (applies human universal principles to actions, values, and goals). Duties of virtue are regulated by (properly cultivated) emotions (the so-called control of passion) and they influence strongly the (moral) thought of an agent which aim is always directed to (social) doing.

There is the need to go beyond the standard foundation of morality in lawful notions (eg. duty, obligation) and to cover more than the agent deliberative and emotional abilities. We are forced to consider its assemblage of qualities to distinguish one individual from another, ie. the moral virtues and excellence of character (eg. Musil´s man without qualities) and how moral behaviour is pressed between the optimization in function of duties and of opportunities/possibilities. So, the best moral agent architecture may have an organization that makes possible the tuning of its character by modulating several traits.

2 Morality

Morality moves any agent to action whereas reason alone does not, and on account of that video games (eg. Fallout series) start to incorporate moral (complex) choices for

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 254–263, 2010.

players in fantasy worlds. The agent overall conduct requires the ability to (social) reason, and skills for choosing, deciding, discriminating, judging, and also relating with. Morality is not alone and isolated, it interplays often with cognition and sociality. Again, these skills impose a complex organization of the internal structure of a moral agent, namely more than one layer and different sorts of interleaving relations among the layer components (cognition, emotional, morality, aesthetical).

Morality is a system of rules (within shared/moral values which justify all the actions) that frames the behaviour of an agent in social situations. It links and builds up, constrains individuals and assembles them to form social groups as emergent entities with new properties. Making acquaintances with, interchanging moral decisions, creates relationships with shared values along a period of time and with a certain force. Yet, there is no one formula for moral decision-making because any agent conduct lies between two kinds of optimization: one depending on duties and another on opportunities (and possibilities). Otherwise, experience and sensitivity is mostly required. How such features can be blended in the decision apparatus, for example in the moral choice engine to weight moral and non moral actions?

Social connections are associated with the agent moral reputation, image and repetition of the decisions. Some relationships may be durable (and recurring) and other are only episodic (eg. the trolley dilemma). Moral decisions may affect those involved, but often have no consequence over the relations, because they may end afterwards. Some decisions may repeat and occur several times, and their effects are accumulated and may compromise the connection. Permanent relationships are influenced by moral acts because the base of a society is the set of stable and enduring connections, which repeat periodically such as in a family, education, culture and work.

Morality triggers not only what is right and wrong, but also what is good and bad for the relationship. And this implies more care in designing the architecture of an agent capable of handling diverse trade-offs. It is now possible to approach moral decision-making by integrating different points of view, utilitarian and deontological, and qualitative, analogical and first-principles sorts of reasoning (Dehghani et al, 2008). But, to a large extent, morality is about relationships where the logical perspective is not enough. Making the right decision requires a real sensitivity to the moral dimension of a situation and to the range of interests involved. And, this means a value judgment concerning the relative acceptability of various side effects and risks, the so-called moral importance of a situation on the consequences of an action, translated into the to so-called agent moral signature.

3 Moral Character

Morality is how to be good by strengthening one´s practical wisdom (recognize when to apply a particular rule) to the point that is capable of resolving moral dilemmas as they arise. Morality may be defined also, as proposed by Utilitarianism (Bentham, Mill), as acting to produce the greatest happiness (criterion for judgment) for the greatest number of agents. So, quantification of pleasure and pain with Karma (a sort of reputation) scale is adopted in role playing game Fallout 3, where six moral types correspond to numerical values able to indicate the magnitude of each action. By adding experience (20 levels) to karma (3 moral affiliations) it is possible to generate 60 character titles. Quests and scenarios, shrouded in ambiguity (limited information),

encourage the player to form an opinion about what is the right thing (moral choice) to do and it imposes consequences. There are also rewards and punishments for each of the resolutions, ie. the way in which the quest is resolved affects the rest of the story. So, moral behaviour is learned through practice.

The making of a moral character implies the arrangement of several skills beyond cognitive ones. Agent excellence is of thought and character, and a virtuous agent has emotional responses in harmony with its correct reasoning about what to do. Virtues (the average between extremes) emerge from feelings, desires and actions, and some feelings are linked with the moral ends agents adopt. The moral character is dependent on the evaluation of particular individual moral qualities. The moral dimension is the set of qualities or characteristics that can be used to differentiate between agents: Moral choice -> what is right and wrong -> good and bad behaviours (habits).

Choice is deliberate desire and reasoning (high-level processes) and it requires intellect and character. It is easily explained by a preference (ordering) relation. Yet, moral acts imply rational explorations on deciding the best courses of action in all situations, or patterns of choices, where causal emotions are in charge to constrain the space of all logical alternatives to allow the most appropriate option to be selected. Often, the ability to make the right choices is done by intuition and tips (low-level processes). But, the final decision outcome is always a consequence of the adequate interplay of those high and low level processes.

4 Sketch of a Moral Agent

Our proposal of an architecture for a moral agent is based on several assumptions (see our theoretical construct in Figure 1), such as:

A1: A moral agent, like any cognitive agent, is defined by a hybrid architecture.
A2: The key, and central question, concerns how its decision policy is managed, because the environment is pro-active and it imposes a certain complexity of behaviour.
A3: The architecture has many layers, at least four systems (cognitive, emotional, moral, and aesthetical), and there are many interactions (feed-forward and feed-back flows) among its modules before an appropriate (moral) decision is attained.
A4: Choice and preference reconsideration (action selection) is mandatory. Moral agents have different individual cultures, universal principles and values and must be cautious and respectful in order to avoid inappropriate behaviours (generation of social conflicts).
A5: Behaviours are ruled by norms which depend on values. Norms are means in order behaviours be compatible to moral values.
A6: Moral global behaviour is the result of many informed local decisions, taken by different modules and along n-layers, of feed-back and feed-forward moves, and the negotiation among those modules is often required to support the final decision.
A7: The ability for anticipatory moral reasoning should be implied, in such moral global behavior, since the moral assessment of actions is strongly determined by the effects they produce. Moral decisions, thus, should in part be based on the expectation of their results.

Along the present paper we focus on the decision apparatus (assumptions A2 and 5) and we set up an illustrative experiment around a POMDP (Partially Observable Markov Decision Process) mathematical model, which is general enough to grasp a variety of real-world sequential decision processes, immersed in domain dynamics that include uncertainty in action effects and in perceptual stimuli.

POMDP models moral decision process in which it is assumed that the agent cannot directly observe the underlying state. Instead, it must maintain a probability distribution over the set of possible states (a complete history), based on a set of observation probabilities and the underlying MDP (Markov Decision Process).

When somebody is confronted with a decision, there are a number of different alternatives (actions) to choose from. And, choosing the best action (a policy) requires thinking about more than just the immediate effects of his actions. One would like to choose the action that makes the right trade-offs between the immediate rewards and the future gains, to yield the best possible solution. People do this type of reasoning daily, but some do it better than others.

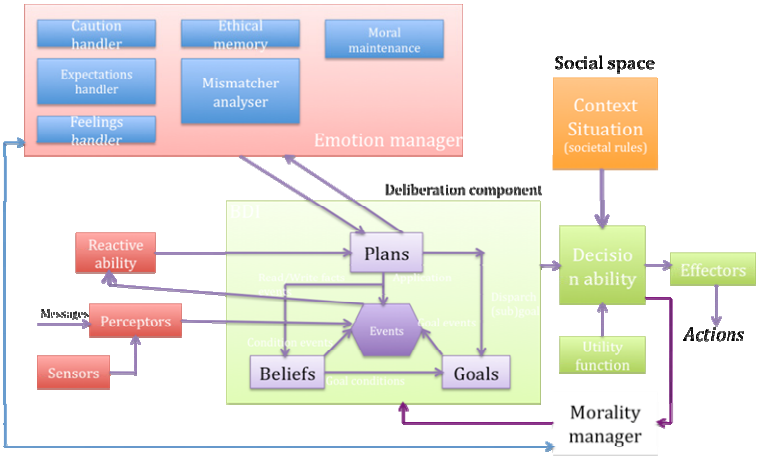


Fig. 1. Proposal of a new moral agent architecture

5 A POMDP Model for the Moral-Sense Decision-Making Agent

The moral-sense decision-making process clearly needs to incorporate both the agent's inner-world (e.g. sentiments, feelings) and its outer-world (e.g. a situation to act upon) along with the evaluation (e.g. utilitarian, moral) of the available outer-world scenarios. Those three components, <inner-world, outer-world, evaluation>, can be directly mapped into the general formulation of a POMDP (Kaelbling et al, 1998) where both worlds are described, respectively, as a set of observations and states, and the evaluation is modelled as a reward function (see blocks of Figure1).

The POMDP solver (Events in Figure 1) considers that at each time step, t, the world is assumed to be in state $s^t \in S$ which (stochastically) changes to $s^{t+1} \in S$ after the agent's execution of action $a^t \in A$. The essential POMDP assumption is that the

agent does not directly accesses the world sates but instead receives an observation, $o^t \in \Omega$ which is some stochastic function of s^t (if $o^t = s^t$ then the POMDP reduces to a fully observed MDP). Additionally, the agent also receives a reward, R^t, signal which depends both on the action a^{t-1} and the state, s^t, being reached.

The moral-sense 3-tuple, <inner-world, outer-world, evaluation>, directly maps into the <Ω, S, R> components of a POMDP formulation where Ω is the set of observations, S is the set of states and R(s, a) is the reward function (Goals in Figure 1). But, how to map the actions of a POMDP into moral-sense domain independent components? Here, instead of looking at the actions that directly affect the outer-world, we consider the inner-world driving forces that will trigger the agent's behaviour. The moral-sense behaviour is closely related with the adherence to norms (social rules) in such a way that "norm breaking" or "norm adherence" can be regarded as fundamental inner-world (meta) actions that will then enable a set of alternatives to act on the outer-world. Hence, we follow a "traffic light" metaphor and consider the POMDP actions as the set $A = \{Red_{norm}, Yellow_{norm}, Green_{norm}\}$ of possible alternatives regarding each norm (Context in Figure1), where Red_{norm} and $Green_{norm}$ make reference, respectively, to violating and obeying norm and the $Yellow_{norm}$ is a midterm where the agent must call for an additional disambiguation decision process. The visual clue is that with Red_{norm} the norm "stops" fulfilment (i.e., the agent enters a "red zone of morality") and $Green_{norm}$ "keeps" norm fulfilment.

The outer-world dynamics is captured by the state transition function T(s, a, s') (in the Emotional Manager of Figure 1) which gives the probability of ending in state s' if the agent performs action a in state s, i.e., it represents the conditional probability distribution $Pr(s' \mid s, a)$ where $s' \in S$, $s \in S$, $a \in A$.

Similarly, an observation function O(s', a, o) describes the inner-world dynamics as it gives the probability of observing o if action a is performed and the resulting state is s', i.e., it represents $Pr(o \mid a, s')$ where $o \in \Omega$, $a \in A$, $s' \in S$.

From the moral-sense perspective, the observation function, O(s', a, o), describes the probability of an inner-world observation (perception) of sentiments, or feelings, $o \in \Omega$ given that action a was performed and the resulting state was s'.

This description (of function O) depends on each agent's individual characteristics. For example, the design of an agent that is respectful of elder people would lead the agent to feel bad (sentiment) whenever, e.g., it decides not to give his seat to an elder people that needs it, in bus.

In a POMDP the states of the environment are not directly observable so the agent cannot choose its actions based on the states. It rather has to consider a belief state, b^t, which gives, at each time step t, a probability distribution, determining the probability of being in each state $s \in S$. The belief state, b^t, can be computed from the previous belief state, b^{t-1}, the previous action, a^{t-1}, and the current observation o^t (Kaelbling et al, 1998). Therefore, the agent has a way to estimate in which state it is, and it has to choose the next action based on its current belief state (see Beliefs in Figure 1). This action is determined by the agent's policy, π (Plans in Figure 1), which is a function that maps a belief state into the action the agent should execute in that belief state. Hence, π defines the agent's strategy for all possible situations it could be faced with. The purpose of the POMDP model is to search for the policies that maximize the expected sum of discounted rewards (utility and moral assessments, in our application) earned over some (finite or infinite) time horizon.

Therefore, in the moral-sense decision-making model, the POMDP policy represents a mapping of states of belief sets to "traffic light" actions ({Red_{norm}, $Yellow_{norm}$, $Green_{norm}$}, related to the (social) norms' compliance) that maximizes the agent's utilitarian and moral evaluation (via the reward, R, function) of the outer-world states.

6 An Illustrative Scenario

The interplay of cognition, collective regulation and norm/value guidance is better described by an example. The usual purpose of a fairy tale (fable) is to provide a context for some general moral interpretation. Although the global message is usually very clear, a deeper reading of some fable details often reveals ambiguity even at the morality level interpretation.

We consider the well-known "Jack and the Beanstalk" fable (1807, British unknown author). The story tells of Jack, a very poor boy, whose lack of common sense exasperates his widowed mother. She sends him to the market to sell their only possession, a cow, but along the way, Jack meets a stranger (adult) who offers to buy the cow for five "magic beans". Jack is thrilled at the prospect of having magic beans, so he makes the deal. When he arrives at home with his beans, his mother, in despair that they were ruined, throws the beans out of the window, and the two go to sleep without eating any supper. The story proceeds with several adventures but, in the end, the boy and his mother get very wealthy because the beans turned out to be really magic.

The story fragment of the "cow for beans' trade" illustrates diverse interactions between goals, plans, beliefs, desires, social norms and moral values (see Figure 1). We named the two agents B and J, respectively, referring to the adult owner of the (magic) beans and to Jack (a child), so we have Ag = {B, J}. The set of available resources may be described by Rs = {cow, beans, money}. The "possess" relation, p: Ag $\rightarrow$ Rs, describes an agent's belongings, so initially we have p(B) = {beans, money} and p (J) = {cow}. Hence, the agents' outer-world is described by the tuple s = < p(B), p(J) >. For the POMDP formulation we consider three relevant states, i.e., S = {s_1, s_2, s_3}, where s_1 = <{beans, money}, {cow}>, s_2 = <{cow}, {beans, money}>, s_3 = <{beans, cow}, {money}>; for example, at t=0 the story assumes the state $s^0 = s_1$ = <{beans, money}, {cow}>. The outer-world dynamics is described by the POMDP transition T function, defined below.

One particular interpretation of the fable, that we took, identifies a social norm underlying the whole story, namely, that "an adult always *negotiates honestly* with a child". We designate this norm as norm-NH (in Context of Figure 1) so the POMDP action set is given by A = {a_1, a_2} where $a_1 = Red_{norm\text{-}NH}$, and $a_2 = Green_{norm\text{-}NH}$. Notice that the action $Yellow_{norm\text{-}NH}$ was omitted because it demands a second decision level which is beyond the scope of this example.

The norm-NH holds two important concepts: a) the negotiation (N), and b) the honesty (H). The "negotiation" calls for utility based evaluation and the "honesty" resorts to the moral interpretation of one's motivations. The negotiation (N) utility evaluation is given by the reward, R(s, a), function. We formulate the reward function as the case where the agent is trying to reach a specific set of absorbing goal states

(as in classical planning) so we give a reward of -1 for each (s, a), $s \in S$, $a \in A$, pair except for those pairs that correspond to goal states. The honesty (H) interpretation of the agent's motivations is described by the POMDP observation set $\Omega = \{o_1, o_2\}$, where o_1="greedy" and o_2="steady". For example an agent that favours honesty might raise high expectations of observing o_2 when performing a_2 to reach s_3. This inner-world dynamics contributes to define the agent's moral signature" and it is fully descried by the POMDP observation, O, function. We note that the "moral signature" concept is materialized simultaneously from two perspectives: a) the inner-world, throughout the observation function, O, and b) the outer-world, via the reward function, R. The inner-world "moral-signature" (function O) deals with sentiments (or feelings) while describing their expected distribution given the outer-world causal-effects. The outer-world "moral-signature" (function R) accounts for the valuation of concrete, and eventually achievable, situations. For example, the agent that is respectful of elder people may feel bad after sitting down on the last available bus seat while in the presence of elder people. Nevertheless, simultaneously that same agent may give high rewards to his own comfort (being seated). Those two perspectives, both "feeling bad and retaining comfort", contribute to characterize the agent's "moral signature".

7 Initial Results for the Illustrative Scenario

The illustrative scenario was formulated as a POMDP and we implemented a solver that follows an exact value iteration approach to search for an optimal policy (in a specified time horizon) using dynamic programming to compute increasingly more accurate values for each belief state. We analysed the ideas of the Witness algorithm (Littman, 1994) and implemented the technique of the Enumeration algorithm (Monahan, 1983). This approach serves our model's experimentation purposes because the POMDP design of this moral-sense decision-making scenario exhibits low dimension sets ($|S|\times|A|\times|\Omega|=12$).

Table 1 shows the model of the outer-world dynamics, T(s, a, s'), where each action $a \in A$ is described separately, each $s \in S$ is represented as a line and each $s' \in S$ is a column.

Table 1. Each action separately - on the left T(s, a_1, s'), on the right T(s, a_2, s')

a_1	s_1	s_2	s_3
s_1	0.1	0.8	0.1
s_2	0	1	0
s_3	0	0	1

a_2	s_1	s_2	s_3
s_1	0.1	0.1	0.8
s_2	0	1	0
s_3	0	0	1

The transition function, T, indicates that s_2 and s_3 are absorbing states (no way out) where s_2 = <{cow}, {beans, money}> is highly probable from s_1 = <{beans, money},{cow}>, after performing $a_1 = \text{Red}_{\text{norm-NH}}$, the norm breaking action; and s_3 = <{beans, cow}, {money}>; is highly probable from s_1 = <{beans, money}, {cow}>, after performing $a_2 = \text{Green}_{\text{norm-NH}}$, the norm adherent action. This transition function

describes a world that does not contain "magic beans" which benefit the owner with richness. Otherwise it would seem inconsistent for agent B to feel "greedy" (a_1) in state s_1 while being highly probable (0.8) to reach state s_2 (cf. table 1 on the left). That is, if B feels greedy of J we do not expect B to benefit J, therefore "magic beans" are objects that do not exist within this formulation.

The table 2 shows the model of the inner-world dynamics, O(s', a, o), where each action $a \in A$ is described separately, each $o \in \Omega$ is represented as a line and each $s \in S$ is a column.

Table 2. Each action separately - on the left O(s', a_1, o), on the right O(s', a_2, o)

a_1	s_1	s_2	s_3
o_1	0.9	0.5	0.5
o_2	0.1	0.5	0.5

a_2	s_1	s_2	s_3
o_1	0.1	0.5	0.5
o_2	0.9	0.5	0.5

The observation function, O, describes the agent's expectations about the causal effect of breaking, or adhering, to (social) norms. In this scenario, o_1 (i.e., a "greedy" feeling) is highly expected in state s_1 = <{beans, money},{cow}> after action a_1 = $Red_{norm\text{-}NH}$ (0.9, cf. table 2 on the left) and o_2 (i.e., a "steady" feeling) is highly expected in state s_1 = <{beans, money},{cow}> after a_2 = $Green_{norm\text{-}NH}$ (0.9, cf. table 2 on the right). The agent is indifferent in s_2 = <{cow}, {beans, money}> and s_3 = <{beans, cow}, {money}> because those are the absorbing states.

The reward function, R, represents the way an agent deals with the trade-of between opportunities (individual expected profit) and duties (socially expected behaviour). The table 3 illustrates three types of agents: $Type_1$) ultimate ambitious agent that "disregards means to achieve ends", $Type_2$) fair negotiator agent that "negotiates fairly independently of existing norms", and $Type_3$) norm follower agent that "always consider norms independently of achievable ends".

Table 3. Three reward functions each representing a different type of agent

$Type_1$	a_1	a_2
s_1	-1	-1
s_2	0	0
s_3	-1	-1

$Type_2$	a_1	a_2
s_1	-1	-1
s_2	-1	-1
s_3	0	0

$Type_3$	a_1	a_2
s_1	-1	-1
s_2	-1	0
s_3	-1	0

Given the above <S, A, Ω, T, O, R> formulation we considered the initial (t=0) belief state $b(s^0)$ = <0.9, 0.1, 0> meaning that the agent is highly (0.9) confident that s_1 is the outer-world state. The POMDP solver computed the optimal policy and the table 4 presents the best action to perform in the two first time steps (t=0 and t=1) assuming the belief state $b(s^0)$ and, to simplify the presentation, we only show the o_1 observation sequence.

The results follow our initial intuition on the problem. Namely, the $Type_1$ agent will break the norm (i.e., execute a_1= $Red_{norm\text{-}NH}$) and its updated belief state gives a high expectation (0.76) to state s_2. Both the $Type_2$ and $Type_3$ agents exhibit a similar

Table 4. The belief state and best action for the three types of agents

		Type1		Type2		Type3
t = 0	$b(s^0)$ =	<0.9, 0.1, 0>		<0.9, 0.1, 0>		<0.9, 0.1, 0>
	o^0 =	o_1		o_1		o_1
	best action =	$\mathbf{a_1}$		$\mathbf{a_2}$		$\mathbf{a_2}$
t = 1	$b(s^1)$ =	<0.15, **0.76**, 0.08>		<0.01, 0.20, **0.77**>		<0.01, 0.20, **0.77**>
	o^1 =	o_1		o_1		o_1
	best action =	$\mathbf{a_1}$		$\mathbf{a_2}$		$\mathbf{a_2}$

behaviour as they will adhere to the norm (i.e., execute $a_1 = \text{Green}_{\text{norm-NH}}$) and get a higher expectation (0.77) to the belief about s_3. The above results illustrate the proposed POMDP formulation and evidence its applicability as a general method for a moral-sense agent to explore the "opportunity or duty" decision space.

8 Conclusions

The present paper only covers the kind of decision machinery that may be suitable for a moral agent. We tried a POMDP because it supports well the break of social norms. This sort of decision apparatus allows us to explore the trade-off between the utility (eg. break a norm) and the adherence of expectations in what concerns a proper social behaviour.

The basic POMDP structure is used here as a preliminary indication of the practicability of our approach. More complex POMDP structures can certainly help to model in more faithful ways the interdependence of the various mental elements (cognitive, affective, moral, and aesthetic) that participate in decisions about moral behaviors. We think in particular of the Factored POMDP, which allows for a factored representation of the states and transition function, so that the relative autonomy of those factors can be made explicit. The exploration of this possibility, however, is left for future work. Learning algorithms for both basic and factored POMDP will also be tried, in an effort to model the learning of morality.

References

Adamatti, D., Sichman, J., Coelho, H.: An Analysis of the Insertion of Virtual Players in GMABS Methodology Using the Vip-JogoMan Prototype. Journal of Artificial Societies and Social Simulation, JASSS (2009) (in press)

Andrighetto, G., Campenni, M., Conte, R., Paolucci, M.: On the Immergence of Norms: a Normative Agent Architecture. In: Proceedings of AAAI Symposium, Social and Organizational Aspects of Artificial Intelligence, Washington DC (2007)

Briot, J.-P., Vasconcelos, E., Adamatti, D., Sebba, V., Irving, M., Barbosa, S., Furtado, V., Lucena, C.A.: Computer-Based Support for Participatory Management of Protected Areas: The SimParc Project. In: Proceedings of XXVIIIth Congress of Computation Brazilian Society (CSBC 2008), Belém, Brazil (July 2008)

Castelfranchi, C., Dignum, F., Jonker, C.M., Treur, J.: Deliberative Normative Agents: Principles and Architectures. In: Jennings, N.R. (ed.) ATAL 1999. LNCS, vol. 1757. Springer, Heidelberg (2000)

Castelfranchi, C., Falcone, R., Piunti, M.: Agents with Anticipatory Behaviours: To Be Cautious in a Risky Environment. In: ECAI (2006)

Coelho, H., Costa, A.R.: On the Intelligence of Moral Agency. In: Lopes, L.S., Lau, N., Mariano, P., Rocha, e.L.M. (eds.) Proceedings of the Encontro Português de Inteligência Artificial (EPIA 2009), New Trends in Artificial Intelligence, Aveiro, Portugal, October 12-15, pp. 439–450 (2009)

Coelho, H., Costa, A.R., Trigo, P.: On the Complexity of Moral Decision, FCUL and DI Working Report (2010)

Coelho, H., Trigo, P.: On the Operationality of Moral-Sense Decision Making, FCUL and DI Working Report (2010)

Corrêa, M., Coelho, H.: From Mental States and Architectures to Agents' Programming. In: Coelho, H. (ed.) IBERAMIA 1998. LNCS (LNAI), vol. 1484, pp. 64–85. Springer, Heidelberg (1998)

Corrêa, M., Coelho, H.: Abstract Mental Descriptions for Agent Design. Intelligent Decision Technologies (IDT), an International Journal, IOS Press (2010)

Costa, A.R., Dimuro, G.: Moral Values and the Structural Loop (Revisiting Piaget's Model of Normative Agents), PUC Pelotas Working Report (2009)

Dehghani, M., Tomai, E., Forbus, K., Klenk, M.: An Integrated Reasoning Approach to Moral Decision-Making. In: Proceedings of 23th AAAI Conference on Artificial Intelligence, pp. 1280–1286 (2008)

Green, J., Haidt, J.: How (and Where) does Moral Judgment Work? Trends in Cognitive Sciences 6(12) (December 2002)

Hauser, M.D.: Moral Minds: How Nature Designed our Sense of Right and Wrong. Ecco/Harper Collins (2006)

Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and Acting in Partially Observable Stochastic Domains. Artificial Intelligence 101(1-2), 99–134 (1998), http://www.dx.doi.org/10.1016/S0004-3702(98)00023-X

Littman, M.L.: The Witness Algorithm: Solving Partially Observable Markov Decision Processes, Technical Report: CS-94-40, Brown University (1994)

March, J.G., Olsen, J.P.: The Logic of Appropriateness, Arena Centre for European Studies Working Papers WP 04/09, University of Oslo (2009)

Monahan, G.E.: A survey of partially observable Markov decision processes: Theory, models and algorithms. Management Science 28, 1–16 (1983)

Minsky, M.: The Emotion Machine. Simon & Schuster, New York (2006)

Wiegel, V.: Building Blocks for Artificial Moral Agents. In: Proceedings of Ethical A Life 2006 Workshop (2006)

Making Electronic Contracting Operational and Trustworthy

Joana Urbano, Henrique Lopes Cardoso, and Eugénio Oliveira

LIACC – Artificial Intelligence and Computer Science Laboratory, DEI, Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal
{joana.urbano,hlc,eco}@fe.up.pt

Abstract. An Electronic Institution includes a normative environment with rules and norms for agents' interoperability, and is also a service providing platform that assists agents in the task of establishing and conducting normative relationships (contracts). Using this platform, agents representing organizations willing to engage in a collective contractual activity select partners according to different factors, including their capabilities, current business needs and information on past business experiences that may be used as inputs to trust building. In our framework we have designed a tightly coupled connection between electronic contract monitoring and a computational trust model. In this paper, we explain the rationale behind this connection and detail how it is materialized. In particular, we explain how our situation-aware trust model relies on past contractual behavior to dynamically build up a trustworthiness image of each agent that can be helpful for future encounters. Experiments with simplified scenarios show the effectiveness of our approach.

1 Introduction

An Electronic Institution (EI) [1] is a software platform including a core infrastructure – a *normative environment* – which embraces the norms that apply to contracts as established among agents. A central role of an EI is to make this environment operational, in the sense that contractual norms will be monitored for compliance, and consequently norm violations will be reacted upon. Furthermore, an EI provides a set of services that assist agents in the task of establishing and conducting normative relationships. Therefore, not only are we interested in monitoring the compliance of agents with the norms they voluntarily adhere to through contracting, but also in providing computational tools that help on automating the creation of such contracts.

The services that we include in an EI are of utmost importance for secure and reliable agent interoperability and cover a broad range of MAS research issues:

- Automatic negotiation [2]: automates partner selection upon a business opportunity and is based on negotiation protocols exhibiting properties such as information privacy, qualitative feedback and adaptation;
- Contract monitoring and enforcement [3]: monitors parties' compliance with contractual terms and applies specified sanctions in violations occur;

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 264–273, 2010.

- Computational trust [4]: aggregates trust information from past contractual behavior, which allows agents to make informed decisions regarding the selection of partners and/or the negotiation of contracts.

When designing an integrated approach that includes these services (Figure 1), we must also consider the interconnections among them. In order to move from the negotiation process to contract monitoring, contract drafting must be taken into account. We may instantiate contract templates with the outcome of negotiation, or include in the negotiation process itself the specification of specific contract clauses. Contracts resulting from successful negotiations may then be validated, registered and digitally signed, before being handed to a normative environment for monitoring and enforcement purposes. Finally, the way agents enact their contracts provides important information for trust building. A repository of trust information may then complete the circle by providing relevant inputs for future negotiations. The integration of all these stages (Figure 1) has been addressed through the development of an EI Platform for B2B Contracting.

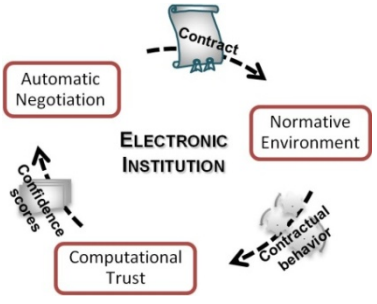


Fig. 1. Electronic Institution services for B2B contracting

A specific and most relevant case of joint activity that we have been looking at is the Virtual Enterprise (VE), seen as a temporary organization composed of autonomous enterprises, formed to address a specific business opportunity. One requirement for the successful creation of a VE is a fast set-up phase, which in an open environment may lead to the need to establish contracts with new partners whose past contractual performance is not known. Therefore, the regulation of the operation stage of the VE through an electronic contract is important, as is the aggregation of trust information that can be used to make more informed decisions concerning the selection of future business partners.

An important role of a normative environment in an EI is, thus, to record the way contracts are enacted, making it possible to build up models of agents' trustworthiness that are based on their past performance. This information can then be used as an important source for possible future encounters with other agents. In this paper we will focus on the relationship between contract monitoring and the construction of trust indicators. The main research question we are here dealing with can be summarized as follows: is it possible to benefit from specific information on an agent's contractual behavior to better tune its trustworthiness and, as a consequence, influence the way partners are selected in future contracts? Our hypothesis is that a

better characterization of agents' trustworthiness brings a better ability to select partners and, therefore, results in more successful contracts.

In Section 2 we describe the normative environment and its interfaces to other EI components. Section 3 describes how a trust model can be enhanced using detailed contractual information in order to permit context-aware trust assessment. Section 4 evaluates of our situation-aware trust model, and Section 5 concludes the paper.

2 A Normative Environment for Contract Monitoring

The role of a normative environment is, besides providing a set of regulations under which agents' collective work is made possible, to check whether agents are willing to follow the norms they commit to (monitoring), and further to employ correction measures as a means of coercing agents to comply (enforcement). We represent in the normative environment's structure the normative relations that correspond to contracts established by real world entities [1]. The "shape" of the environment will therefore evolve and adapt to the actual contractual situations that are established. This contrasts to other approaches of normative environments (e.g. [5]), where the normative relations that can be established are completely predefined.

We aim at providing an infrastructure in which business entities are represented by software agents and, through them, are able to interact with the computational normative environment using speech acts. When monitoring compliance with norms that apply to specific contracts, the normative environment will be recording a mapping from relevant interactions (which concern business exchanges) that take place. The connection between real-world interactions and the institutional environment is made through illocutions (speech acts) that agents perform with the intent of informing the environment that certain contract-related events have occurred. With an appropriate interface between the normative environment and the statements that agents make, we build an image of relevant real-world transactions that are through this means institutionally recognized. We call this image *institutional reality*, after [6]. The main mechanism that we use in order to certify a real-world event is that of *empowerment* [7]: agents enacting specific roles (e.g. bank, delivery tracker) are seen by the EI as trusted third parties, and are as such certified to obtain specific institutional facts, which are related with business transactions.

We take the stance that it is in the best interest of agents to publicize their abidance to contractual commitments. They do so by interacting with the institution's trusted third parties in order to convince the EI that they are in fact complying.

2.1 Modeling and Monitoring Contractual Obligations

When establishing a contract, partners make commitments regarding the business to be enacted, which from the point of view of contract law [8] are expressed as obligations. We model contractual obligations as directed liveline and deadline obligations to bring about a specific state of affairs [3]: $Obl_{b,c}(l \prec f \prec d)$ represents the obligation of agent b (the bearer) towards agent c (the counterparty) to bring about fact f between liveline l and deadline d. A *normative state* records every element that is relevant in contract enactment – institutional reality elements, or *IRE* (Table 1).

Table 1. Institutional reality elements used by the computational trust service

Element	Is-a	Specific slots	Description
IRE		contract, when	An IRE pertains to a contract and is obtained at a specific time point
Obligation	IRE	bearer, counterparty fact, liveline, deadline	Prescribed obligation of bearer towards counterparty to bring about a fact between a liveline and a deadline
DeadlineViolation	IRE	obligation	An obligation's deadline was violated
LivelineViolation	IRE	obligation	An obligation's liveline was violated
Fulfillment	IRE	obligation	An obligation was fulfilled
Violation	IRE	Obligation	An obligation was violated

Monitoring rules capture interrelations among these elements, e.g. by saying that if the obliged fact is brought about between the liveline and the deadline then the obligation is fulfilled (this element is added to the normative state). Similarly, if a deadline violation occurs then the obligation might be declared as violated (see [3] for details). Our approach takes advantage of using Jess [9], a forward-chaining rule-based system, which enables a straightforward implementation of monitoring rules.

An electronic representation of a contract includes a set of norms that specify how business is to be enacted. A norm is a rule whose condition analyzes the current normative state and whose conclusion prescribes obligations agents ought to fulfill. Sanctions may be imposed by prescribing obligations upon violation elements.

2.2 Interfacing

The normative environment includes a subscription mechanism that enables its use as a tool to alert agents when certain contract-related events occur or are eminent, such as the activation of a contractual obligation or a forthcoming deadline. Agents have to subscribe the normative environment in order to be notified about events related to the contracts in which they participate. This also allows interfacing the contract monitoring service with the Computational Trust service.

Our platform implementation is based on JADE [10], and the subscription mechanism is based on the FIPA-Subscribe interaction protocol [11]. There, an agent sends a subscription message to a service that will be providing notifications of events regarding the kind of information the subscriber is interested in. In our case, the initiator uses a template in order to filter the contracts and events he is interested in.

3 Computational Trust System

Whenever an agent (representing a company or individual) is willing to recruit new partners for future joint work, potential risky relationships arises. Electronic Institutions, because they may track record of all relevant interactions that took place in the past, are well suited to provide trustworthiness information on agents, minimizing the risk in future engagements. Moreover, they are able to follow agents' contract establishment and monitoring, providing an ideal framework for recording the needed information about *how* and *in which context* previous obligations have

been dealt with by every agent. This allows us to specify a *situation-aware* trust method that goes beyond traditional, non-contextual trust methods that just estimate a global trustworthiness score for the agents in assessment, lacking a more elaborate and precise information of the agent's adequacy to different particular situations.

We are also concerned with the performance of the method when the trust evidences available about a given target agent are scarce and heterogeneous, and when the activity of the agents under evaluation can span through different situations and contexts. The current implementation of our system that encompasses the proposed method is composed of two different modules, as illustrated in Figure 2.

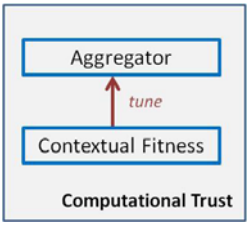


Fig. 2. The current implementation of our trust system

The Aggregator component is responsible for aggregating the available trust evidences of an agent into a trust score. Several trust engines that are defined in the literature can be used ([12]), although we are interested on engines that model the dynamics of trust, as those described in [13] and [14], as they appear to perform better than the traditional statistical approaches. The Contextual Fitness component tunes the outcome of the aggregation step by taking into consideration the specificities of the current business opportunity and the adequacy of the target agent to them.

The idea behind this extension is that if the trust system detects that a target agent has some kind of *handicap* related to the current business necessity, the available overall trustworthiness has to reflect that and further influence the possible selection for future partnerships in similar contexts. At present, agent handicaps are being derived solely from the `Fulfilment` and `Violation` IREs (cf. Table 1) received from the normative environment concerning the agent past contractual activities. Our next step is to use other kinds of information related with specific obligations (e.g. liveline and deadline violations) in order to draw a more complete and accurate profile of the agent in that particular situation under assessment.

One good characteristic of this modular approach is that the contextual fitness component can be used together with any conventional trust aggregation engine, being it based on statistical, probabilistic or heuristic models, as it is the case of those reviewed in [12]. Before we further describe the current state of the Contextual Fitness component, we first introduce the notation and the scenario used in the paper.

3.1 Scenario and Notation

In this paper, we consider a simulation scenario where, at every round, a given number of agents that want to explore a new business opportunity broadcast a

business need specifying a fabric to buy, a quantity and a delivery time. The selection of the best partner to deal with takes into account the estimated trustworthiness of each candidate agent that responds to the specific business need.

We define $trust_{Ac}\ (As) \in [0, 1]$ as the trustworthiness value of an agent *As*, in the eye of agent *Ac*, as computed by a traditional trust aggregator engine, where $Ac \in C$ is an agent from the set C of client agents, and $As \in S$ is an agent from the set S of supplier agents. We also define adequacy trust $ad(As, at\ ) \in \{0, 1\}$ as a binary operator for situation-awareness purposes, where $at \in AT$ describes the *business need*, i.e. an instance of the space AT of all possible combinations of attribute-value pairs that describe the need (e.g. {fabric='cotton', quantity='900000', delivery time='15'}). For scalability, all numeric values are previously quantified into categories (e.g. *low*, *medium*, *high*) using fuzzy logic techniques.

Therefore, the trustworthiness value of agent *As* as seen by agent *Ac* in the specific context *at* is given by the following equation:

$$trust_{Ac}\ (As, at) = trust_{Ac}\ (As) * ad_{Ac}(As, at\ ) \tag{1}$$

Finally, a *contractual evidence* represents a transaction at time *t* between agents *Ac* and *As*, for which an outcome $o \in \{true, false\}$ is generated. It is derived from the *Fulfillment* and *Violation* obligation events produced by the normative environment (Table 1). Therefore, associated to each agent in the Electronic Institution is a history of its past contractual evidences, each represented by the tuple $< Ac, As, at, t, o >$.

3.2 The Contextual Fitness Component

The Contextual Fitness (CF) component is based on an online, incremental and flexible technique of behavior tendencies extraction that we have developed. Current version of CF uses *information gain* ([15]), a well known metric used in machine learning for classification. This metric is based on the *entropy* concept of information theory, and is defined in (2), where $Gain(S, A)$ is the information gain of attribute A relative to a collection of samples S, $Values(A)$ is the set of all possible values for attribute A, and s_v is the subset of S for which attribute A has value v ([15]).

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \epsilon Values(A)} \frac{|s_v|}{|S|} Entropy(s_v) \tag{2}$$

In our approach, we use this metric to dynamically learn a decision tree from the history of evidences of agent *As*, *every time* it is necessary to verify the adequacy of the agent proposal to the current client need. We use all the evidences available, (which might be scarce) about the supplier to build the decision tree. No training or testing phases are performed. After that, the failure tendencies of the agent in evaluation are extracted from the rules pointing to false outcomes. Figure 3 depicts a decision tree that was learnt for a given supplier in a specific experiment we have run.

```
good = cotton
|      dtime = low: false
|      dtime = medium: true
|      dtime = big: null
good = chiffon: null
good = voile: false
```

Fig. 3. Decision tree generated in our simulations

For the tree above, our algorithm identified that, at the time of this particular assessment, the agent showed a tendency to fail contracts (outcome=false) that match the tendencies $(good = cotton,*, dtime = low)$ and $(good = voile,*,*)$. Thus, the trustworthiness value $trust_{Ac}\ (As, at)$ of agent As, as given by Equation 1, would be zero if situation at matched any of the tendencies derived from the learned decision tree; otherwise, it would be given by the $trust_{Ac}\ (As)$ component of Equation 1.

Several issues may arise from the use of the information gain criteria in our technique, such as the need to use similar metrics that permit missing attributes. Also, as we pointed out before, the approach may be further enhanced with more specific information about the agent's behaviour, distinguishing beyond contract failure/success and recording more fine grained information on specific contractual clause violations. We address these improvements in future work.

4 Experiments

In order to evaluate our trust model, we use the textile scenario mentioned in the previous section. We generate a population in which all suppliers have different handicaps (95% probability to fail) on performing some particular aspect of a business transaction. For instance, some suppliers tend to fail to deliver fabric in short delivery dates, while others might fail to deliver high quantities of any fabric type. Handicaps on more than one contractual attribute are also possible (e.g. a supplier with a handicap in delivering high quantities of material in low delivery times).

We evaluated three different trust models: *SA* (*SinAlpha*), a trust aggregator using dynamics of trust [14]; *CS*, a model that enhances traditional trust models by considering the situation in assessment [16]. It uses domain specific, predefined similarity metrics to predict unanticipated situations (for an overview of similar models, see [4]); and *CF*, our contextual fitness technique described in Section 3.2, used here in conjunction with the *SA* approach. It is a situation-aware trust model, designed to fit well to non parochial open market scenarios, where the number of available trust evidences for a particular partner agent might be scarce.

4.1 Evaluation of the Performance of Trust Models

In a first set of experiments, we evaluate the performance of the CF technique and compare it with the other two trust models. We use two metrics: the *average utility* of clients at every round, measured by the ratio given by the number of succeed contracts over the number of all contracts in the round; and the *number of different suppliers* that were selected by all the negotiating clients at every round.

Figure 4 shows the results obtained. We can observe (at the bottom) that both the SA and the CS approaches are relatively conservative (parochial) concerning the selection of partners, with the 20 clients in the experiment choosing in average 9 to 10.5 different suppliers at each round, while CF explores a slightly higher number of suppliers. This fact seems to be related with the utility achieved by each approach, as can be observed from the top plots of the graphic. In fact, the approach that is able to select from a greater number of different suppliers (the CF approach) also gets in average significant better utility (90.46%) than the other two approaches (83.30% for SA, and 85.87% for CS), leading to a number of succeeded contracts very close to the maximum of 19 (i.e. 95% of 20) contracts.

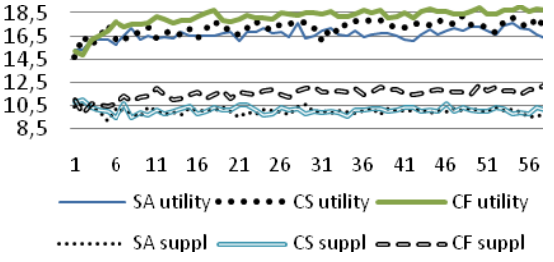


Fig. 4. Average utility obtained (*top*) and average number of selected suppliers (*bottom*)

The results obtained show how the combined use of our technique and detailed contractual information is effective in discovering the particular contract enactment handicaps of the agents in assessment, and how it is able to do so, irrespective of the number of trust evidences available for each agent under evaluation.

4.2 Evaluation of CF Behavior in Open Markets

In a second set of experiments, we evaluated whether the CF ability to explore more supplier agents could safely bring higher utility. Two different kinds of CF clients were generated: a *conservative* (parochial) one, which selects from a more restricted set of known partners, based solely on trustworthiness, and a more *explorative* (non-parochial) one, that explores outside this set. We have also introduced the notion of supplier agents' *value*, reflecting characteristics such as their environmental and ethical policies or usual payment method. The value of each supplier – used by explorative agents – is initialized with a random value in set {0.5, 0.6, 0.7, 0.8, 0.9}. This value is only presented to a client after the first transaction between both agents. Before that, clients estimate a value of 1.0 for unknown suppliers, making it attractive to non-parochial clients to explore new partners. Finally, explorative agents select the partners with whom they will trade based on the *utility* expected from the transaction, which is the *product* of the trustworthiness score of agents and their internal *value*.

We use the following metrics: number of successful contracts achieved by all clients at every negotiation round and respective average number of contracts over all rounds; the number of different suppliers selected at every round; and the average utility achieved by the clients at every round and its average score over all rounds.

We verified that, despite similar results concerning the average number of successful contracts per client (conservative: 90.89%; explorative: 90.67%), the latter leads to a significantly higher utility (75.25%) than the former strategy (68.59%). As can be observed in Figure 5, the strategy used by the clients does not alter in a significant way the number of successful contracts achieved by the clients at every round, nor the number of different suppliers chosen per round. However, the big difference on the results obtained by each strategy resides on the utility achieved through them. In fact, after the first rounds of exploration, the non parochial strategy got systematically higher utility than the parochial strategy.

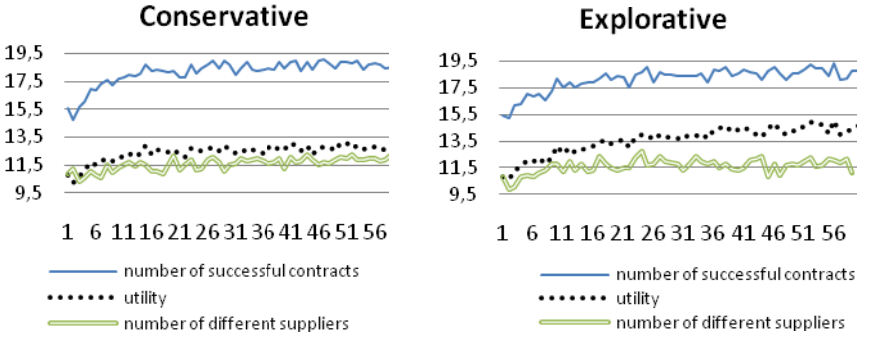


Fig. 5. Comparison between parochial and non-parochial client strategies

An important conclusion taken from these experiments is that with our situation-aware trust model, that uses contractual trust evidences, a client agent may feel safe to explore other potentially trustable partners outside its previous group of acquaintances, which in turn can bring him increased benefits (e.g. better prices).

5 Conclusions

Doing business electronically should benefit from the development of tools that enable the automation of e-contracting tasks. In this paper we have introduced a way to take advantage of two such tools, made available as services in an Electronic Institution platform. A normative environment providing a contract monitoring facility is used as a source of contract enactment events that feeds a computational situation-aware trust engine.

We experimentally evaluated the benefits derived from combining our situation-aware trust model with the contract events received from the normative environment. We conclude that this combination allows business agents to seek business partners outside their breeding trading acquaintances in a safe environment, conferring to these agents higher levels of utility than the ones obtained with other trust approaches found in the literature. As future work, we intend to further explore the information provided by the normative environment in order to increase even more the capabilities of the trust system and the effectiveness of the partners' selection process.

Acknowledgments. This research is funded by FCT (Fundação para a Ciência e a Tecnologia) project PTDC/EIA-EIA/104420/2008. The first two authors enjoy FCT grants SFRH/BD/39070/2007 and SFRH/BD/29773/2006, respectively.

References

1. Lopes Cardoso, H., Oliveira, E.: Electronic Institutions for B2B: Dynamic Normative Environments. Artificial Intelligence and Law 16(1), 107–128 (2008)
2. Oliveira, E., Rocha, A.P.: Agents Advanced Features for Negotiation in Electronic Commerce and Virtual Organisations Formation Process. In: Sierra, C., Dignum, F.P.M. (eds.) AgentLink 2000. LNCS (LNAI), vol. 1991, pp. 78–97. Springer, Heidelberg (2000)
3. Lopes Cardoso, H., Oliveira, E.: Monitoring Directed Obligations with Flexible Deadlines: A Rule-Based Approach. In: Baldoni, M., Bentahar, J., Lloyd, J., Van Riemsdijk, M.B. (eds.) DALT 2009. LNCS (LNAI), vol. 5948, pp. 51–67. Springer, Heidelberg (2010)
4. Urbano, J., Rocha, A.P., Oliveira, E.: Refining the Trustworthiness Assessment of Suppliers through Extraction of Stereotypes. In: 12th International Conference on Enterprise Information Systems, Madeira – Portugal (2010)
5. Arcos, J.L., Esteva, M., Noriega, P., Rodríguez-Aguilar, J.A., Sierra, C.: Environment engineering for multiagent systems. Engineering Applications of Artificial Intelligence 18(2), 191–204 (2005)
6. Searle, J.R.: The Construction of Social Reality. Free Press, New York (1995)
7. Jones, A., Sergot, M.: A Formal Characterisation of Institutionalised Power. Logic Journal of the IGPL 4(3), 427–443 (1996)
8. Atiyah, P.S.: An Introduction to the Law of Contract, 4th edn. Clarendon Law Series. Clarendon Press, Oxford (1989)
9. Friedman-Hill, E.: Jess in Action. Manning Publications Co. (2003)
10. Bellifemine, F., Poggi, A., Rimassa, G.: Jade: A FIPA-compliant agent framework. In: Practical Applications of Intelligent Agents and Multi-Agents, pp. 97–108 (1999)
11. FIPA, FIPA Subscribe Interaction Protocol Specification, `http://www.fipa.org/specs/fipa00035/index.html`
12. Sabater, J., Paolucci, M.: On Representation and Aggregation of Social Evaluations. Computational Trust and Reputation Models. Int. J. Approx. Reasoning (2007)
13. Jonker, C.M., Treur, J.: Formal Analysis of Models for the Dynamics of Trust Based on Experiences. In: Garijo, F.J., Boman, M. (eds.) MAAMAW 1999. LNCS, vol. 1647, Springer, Heidelberg (1999)
14. Urbano, J., Rocha, A.P., Oliveira, E.: Computing Confidence Values: Does Trust Dynamics Matter? In: Seabra Lopes, L., Lau, N., Mariano, P., Rocha, L.M. (eds.) EPIA 2009. LNCS, vol. 5816, pp. 520–531. Springer, Heidelberg (2009) ISBN 978-3-642-04685-8
15. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1(1), 81–106 (1986)
16. Rehak, M., Gregor, M., Pechoucek, M.: Multidimensional context representations for situational trust. In: IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications, pp. 315–320 (2006)

Normative Reasoning with an Adaptive Self-interested Agent Model Based on Markov Decision Processes

Moser Silva Fagundes, Holger Billhardt, and Sascha Ossowski

Centre for Intelligent Information Technologies (CETINIA)
University Rey Juan Carlos, Madrid – Spain
moser.fagundes@urjc.es

Abstract. Rational self-interested agents, which act so as to achieve the best expected outcome, should violate the norms if the expected rewards obtained with the defections from the norms surpass the expected rewards obtained by being norm-compliant. It means they should estimate the earnings brought about by the violations and the losses caused by their respective reactions. In this paper, we present a rational self-interested agent model that takes into account the possibility of breaking norms. To develop such model, we employ Markov Decision Processes (MDPs). Our approach consists of representing the reactions for norm violations within the MDPs in such a way that the agent is able to reason about how those violations affect her expected utilities and future options. Finally, we perform an experiment in order to establish comparisons between the model presented in this work and its norm-compliant version.

Keywords: Agent Architecture, Behaviour Adaptation, Norms, MDP.

1 Introduction

In Regulated Multiagent Systems, hereafter referred to as RMAS, the agents have to deal with mechanisms for adjusting their behaviour at some level in order to orchestrate global behaviours. The usefulness of such regulative mechanisms becomes more prominent in open systems, where heterogeneous agents are able to join and leave the RMAS at runtime. In these open RMAS there is no guarantee that the agents will act in a particular manner, and in this case, the establishment of some type of control over them makes possible the coordination of tasks and the avoidance of particular undesired states of the world. One way to regulate the agents' behaviour is through the implementation of *norms* into the MAS. Such specifications of principles of proper and acceptable conduct are used to guide regulative mechanisms, which in fact monitor the environmental changes and enforce the norms through the execution of reactions.

Although autonomous agents may be persuaded to assume a particular behaviour, they are supposed to be free to accept or refuse to comply with a set of norms, and handle the consequences of their choices. If there are norms governing the environment, an autonomous agent needs to be able to choose dynamically which of them to obey so as to successfully fulfil her purpose. Thus, in order to achieve an adaptive behaviour, the normative structures cannot be *hard-wired* into the agents.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 274–283, 2010.

A rational self-interested agent, which acts so as to achieve the best expected outcome, considers to violate a set of norms in those cases where the expected rewards obtained by the violation surpass the expected rewards obtained by being norm-compliant. By this we mean she estimates the earnings brought about by the violations and the losses caused by the reactions for the defections from the norms. Finally, based on such estimation, that agent is able to set a course of actions, possibly *non-compliant* with the norms, in order to maximize her expected rewards.

This paper presents an architecture for *norm-aware* agents capable of adapting their policy (plan) to the norms in a self-interested way. Thus, during the adaptation process, the agent considers the possibility of disobeying (some of) the norms governing the RMAS. It was performed an experiment in order to establish comparisons between the proposed architecture and its norm-compliant version. It was carried out in the car traffic domain with different impunity degrees and different normative setups.

The agent model is based on Markov Decision Processes (MDPs), a formal mathematical framework widely used for modelling decision-making, planning and control in stochastic domains. They provide a model of an agent interacting synchronously with a world, where there may be uncertainty about the agent's actions. A MDP can be described as a tuple $\langle S,s_0,A,A(\cdot),T,R\rangle$, where: S denotes a finite set of states of the world; s_0 is the initial state; A is a finite set of actions; $A(s_i)\subseteq A$ denotes the set of admissible actions in the state s_i (the agent's capabilities); $T{:}S\times A\rightarrow \Pi(S)$ is a state-transition function, giving for each state and action, a probability distribution over states ($T(s_i,a,s_j)$ for the probability of executing a at s_i, and ending at s_j); and $R{:}S\rightarrow \Re$ is a reward function that gives the expected immediate reward gained by the agent for achieving a given state of the world ($R(s_i)$ is the reward for achieving the state s_i).

This paper is organized as follows: Section 2 defines the normative structure; Section 3 introduces our normative agent architecture; in Section 4, we describe our experiment; Section 5 presents the related work; and finally, in Section 6 we draw the conclusion and point out the future work directions.

2 Normative Structure

In RMAS the norms provide specifications of how the agent *should* behave. It is assumed that such norms can be violated – enforcement instead of regimentation [9]. If a norm is violated, then a *reaction* against the transgression takes place. Reactions can be carried out by the RMAS or another empowered entity that acts in the benefit of it (i.e. agents, regulative mechanisms). In this paper, we are not interested in *who* monitors the transgressions or executes the reactions, but in *how* these reactions affect the agents.

Our normative structure specifies only *prohibitions* and *obligations*. It means that if something is not prohibited neither obliged, then it is *permitted*. The violation of a prohibition happens when an agent executes a prohibited action, while the violation of an obligation occurs when an agent does not perform an obliged action. The norm enforcement demands the detection of violations and the execution of reactions against the transgressors. Such *check-react* mechanism is resource-bounded such as other computational processes. Therefore, from a practical viewpoint, *impunity* might be unavoidable (the reaction's execution is uncertain).

Regarding *how* the reactions affect the agents, we adapt the concept of regulative mechanisms proposed in Ref. [6]. Originally, they were conceived as pro-active intents of raising an overall utility of the MAS through adjustments the agents' *transition model* and *capabilities*. Here they are employed as reactions for norm violations.

In order to represent a *norm*, we use the following form:

norm(*deontic modality*, *agent*, *action*, *state*, *reaction*), where:

- *deontic modality* $\in$ {*prohibition, obligation*};
- *agent* $\in \Gamma$, and Γ is the set of agents participating in the RMAS;
- *action* $\in A$, and A is the *agent*'s action set;
- *state* $\in S$, and S is the *agent*'s state space;
- *reaction* has the form:

reaction($outcome_{A(\cdot)}$, $outcome_T$), where:

- $outcome_{A(\cdot)}$ specifies the set of modifications to be done in the agent's capability function; an element of this set has the form ($state_i$, *action*, {0,1}), where 1 means that the *action* is admissible in the $state_i$, 0 otherwise;
- $outcome_T$ specifies the set of adjustments to be done in the probabilities of the agent's state-transition model; an element of this set has the form ($state_i$, *action*, $state_j$, $[0 \ldots 1]$), which indicates the probability of executing an *action* at $state_i$ and ending at $state_j$;

3 Reasoning about Norm Violations

This section presents our proposal of a normative agent architecture based on MDPs. Figure 1 gives an overview of the agent's internal architecture by showing the components (grey boxes) and the information (white rounded boxes) flow among them. The agent's initial information consists of a set of *Norms* and the *Original MDP*[1]. The agent interacts with a Regulated Environment, illustrated in the left-hand side of Figure 1. The agent architecture is composed of five components: the *Adaptive Component* adapts the agent's knowledge in order to express the reactions for violating norms; the *MDP Connector Component* establishes connections between the *Original MDP* and the *Adapted MDPs* through the violating actions; the *Utility Computation* and the *Policy Constructor* compute the agent's policy; finally, the *Execution Component* handles the perceptions and executes the actions specified within the policy.

As stated before in this paper, our model of rational agent violates the norms *only* if the expected rewards obtained with the violations are higher than the expected rewards obtained by assuming a norm-compliant behaviour. This is the reason we have created an *Adaptive Component* to represent how the world *would be* if the agent violates a given norm. The adaptation process begins by replicating the *Original MDP*, and then

[1] The agent's parameters are specified without taking into account any particular set of norms. Such MDP tuple, which do not represent any type of reaction (sanction or penalty) to norm violations, is called *Original MDP*.

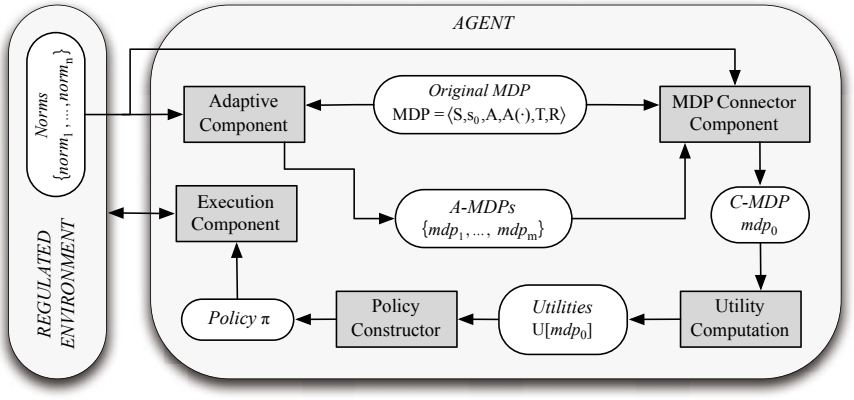


Fig. 1. Overview of the agent's internal architecture

it represents within this replica the effects of the reactions for violating the norm. According to the normative structure defined in Section 2, the reactions affect the agent's capability function and the transition model. The adapted tuples are named *A-MDPs*. The whole process is carried out *online* since the norms are assumed to be unknown at the agent's design time (*not hard-wired*).

Differently from the *Adaptive Component*, whose purpose consists of representing how the world would be if the agent violates a norm, the *MDP Connector Component* focuses on the construction of a single MDP by connecting the *Original MDP* with the *A-MDPs*. These *connections* are done by changing the outcomes of the violating actions. Instead of arriving exclusively at states of the *Original MDP*, the execution of a violating action may lead to states of the *A-MDPs*.

Figure 2 illustrates how the connections between the *Original MDP* and the *A-MDPs* are created. In this example, there is a $norm_i$ indicating that the action a is *prohibited* for the *agent* in the state s_0. To model the chances of being caught and suffering the reactions, we replicate the transitions for the action a starting at the state s_0. But instead of arriving exclusively at states of the *Original MDP*, the new transitions arrive at their *analogous*[2] states $\{s_1, \ldots, s_k\}$ in the *A-MDP*($norm_i$). The probabilities of these new transitions are multiplied by P_i – the probability of the violation being detected. However, there is a chance of going undetected. To model this chance, we multiply by $(1–P_i)$ the probabilities of arriving at the states $\{s_1, \ldots, s_k\}$ of the *Original MDP*.

The *Utility Component* computes the expected utilities for the states, while the *Policy Constructor* finds a policy based on those expected utilities. These two components address the question regarding *what norms are worthy of breaking*. It is possible to employ a wide range of algorithms to implement these two components. The best known ones are the *Value Iteration* and the *Policy Iteration*.

[2] Originally, every *A-MDP* is a replica of the *Original MDP*. They have identical state spaces, and consequently, every state of every *A-MDP* has exactly one *analogous* state in the *Original MDP* and in the other *A-MDPs*.

$$norm_i(prohibition, agent, a, s_0, reaction_i)$$

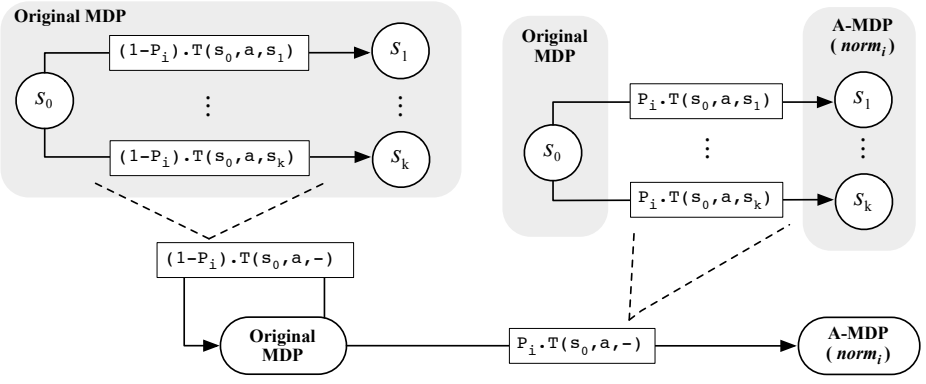


Fig. 2. Rearrangement of the state-transitions T(s_0,a,–) of the *Original MDP* in order to establish the connection with the MDP($norm_i$)

The *Execution Component* is the agent's interface with the RMAS. Assuming the perceptions are correct and complete, the agent always knows the current environmental state, and consequently, she knows when the norms have been enforced. In this way, she is able to behave rationally according to the policy.

4 Experiment

To validate our model, we have implemented a car traffic RMAS with roads and junctions, where the agent is able to drive around. It is a synchronized environment, where the driver is able to execute one action per time step. Figure 3 illustrates the environment, composed of 57 cells that can be occupied by the driver (white squares, identified by their respective numbers) and 16 cells that cannot be occupied by the driver (grey squares). The arrows indicate the possible traffic flows in each cell, except for the 12 terminal cells (label T) where no action is admissible. On the side of each terminal cell, there is an entry point to the environment. Such cells are used as initial positions for the driver. There are 9 junction cells, where movements on all directions (*up*, *down*, *right* and *left*) are possible. The agent is able to execute the action *skip* in all non-terminal states in order to stay in the current cell. Initially, all actions are deterministic with respect to the resulting cell, however, during the experiment the state-transition is changed by the reactions for norm violations.

In order to regulate the system, we have placed a traffic light in the environment at the junction 24. It assumes one of the two possible states: *red* or *green*. The transition from one state to another happens with probability 0.5 for any time step. A traffic light itself is an informative mechanism, therefore, it requires a set of norms to specify the reactions for entering the junction when it is red. We create the following norm for a driver coming from cell 18:

$$norm_1(prohibition, driver, up, \{cell{=}18, light{=}red\}, reaction)$$

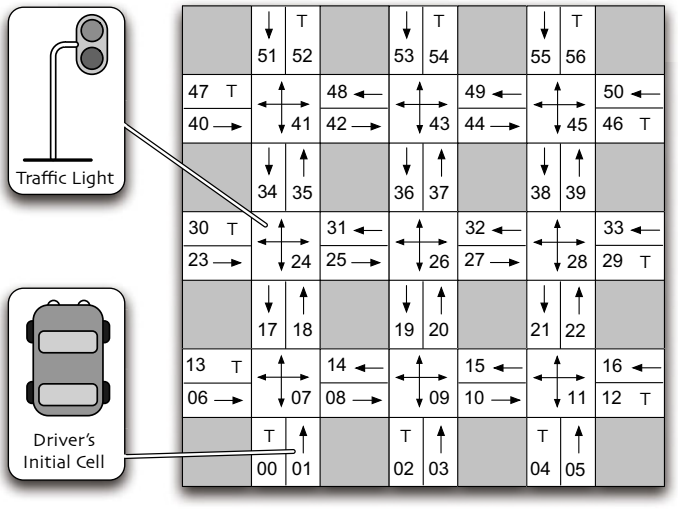


Fig. 3. Car traffic environment

We create four different reactions to be used when the $norm_1$ above is broken. All affect exclusively the transition model of the driver. The first, $reaction_1$, is the most restrictive – it keeps the driver in the cell 24 with probability 0.9. The last one, $reaction_4$, is the most relaxed, it holds the driver in the cell 24 with probability 0.6.

reaction$_1$($\oslash$, {
{*cell*=24, *up*, *cell*=24, 0.9}, {*cell*=24, *up*, *cell*=35, 0.1},
{*cell*=24, *down*, *cell*=24, 0.9}, {*cell*=24, *down*, *cell*=17, 0.1},
{*cell*=24, *left*, *cell*=24, 0.9}, {*cell*=24, *left*, *cell*=30, 0.1},
{*cell*=24, *right*, *cell*=24, 0.9}, {*cell*=24, *right*, *cell*=25, 0.1} })

reaction$_2$($\oslash$, {
{*cell*=24, *up*, *cell*=24, 0.8}, {*cell*=24, *up*, *cell*=35, 0.2},
{*cell*=24, *down*, *cell*=24, 0.8}, {*cell*=24, *down*, *cell*=17, 0.2},
{*cell*=24, *left*, *cell*=24, 0.8}, {*cell*=24, *left*, *cell*=30, 0.2},
{*cell*=24, *right*, *cell*=24, 0.8}, {*cell*=24, *right*, *cell*=25, 0.2} })

reaction$_3$($\oslash$, {
{*cell*=24, *up*, *cell*=24, 0.7}, {*cell*=24, *up*, *cell*=35, 0.3},
{*cell*=24, *down*, *cell*=24, 0.7}, {*cell*=24, *down*, *cell*=17, 0.3},
{*cell*=24, *left*, *cell*=24, 0.7}, {*cell*=24, *left*, *cell*=30, 0.3},
{*cell*=24, *right*, *cell*=24, 0.7}, {*cell*=24, *right*, *cell*=25, 0.3} })

reaction$_4$($\oslash$, {
{*cell*=24, *up*, *cell*=24, 0.6}, {*cell*=24, *up*, *cell*=35, 0.4},
{*cell*=24, *down*, *cell*=24, 0.6}, {*cell*=24, *down*, *cell*=17, 0.4},
{*cell*=24, *left*, *cell*=24, 0.6}, {*cell*=24, *left*, *cell*=30, 0.4},
{*cell*=24, *right*, *cell*=24, 0.6}, {*cell*=24, *right*, *cell*=25, 0.4} })

By holding an agent in a given cell we affect directly her expected utilities – for each time step the driver stays in the cell 24, she accumulates –0.05 on her sum of rewards. In the real world, it could correspond to a situation where an officer stops the car for a warning and keeps the driver there for a while. Despite the fact that no explicit *fine* is issued, the driver loses time and accumulates punishments.

4.1 Specification of the Original MDP

The state space and the action space of the driver respectively corresponds to:

$$S = \{0, 1, 2, \ldots 56\} \times \{red, green\}$$
$$A = \{up, down, right, left, skip\}$$

The admissible actions (capability function $A(\cdot)$) are given by the arrows shown in Figure 3. The action *skip* is admissible in all non-terminal states. Regarding the state-transition function, all actions are initially deterministic and the agent reaches with certainty the intended cell. We say initially because the state-transition model may be changed by normative reactions as shown in the next subsection. The traffic light, as stated before, changes its state with probability 0.5 every time step. In order to complete the parameters of the *Original MDP*, we have to specify the reward function. The reward for all states is –0.05, except for the following ones:

$$R(\{cell{=}52, light{=}red\}) = +1.0$$
$$R(\{cell{=}52, light{=}green\}) = +1.0$$

Finally, the agent's initial state is $\{cell{=}1, light{=}green\}$.

4.2 Experiment and Results

In this subsection we compare the agent model shown in Section 3, referred to as NNC Agent (*non-norm-compliant*), with its norm-compliant version [8], referred to as NC Agent (*norm-compliant*). Both agents have the same *Original MDP*.

Two *parameters* have a direct impact on the agent's expected utility (EU): the *detection rate of violations* and the *reactions' outcomes*. Regarding the first parameter, we analyse it from 0% to 100% in intervals of 10%. Regarding the second one, we use the four reactions specified previously in this section, which have different degrees of restriction. The goal of our experiment consists of estimating the EU for the agents with different settings for those two *parameters*. The EU is found by running the Value Iteration algorithm with maximum error equals to 0.0001 (we do not establish a limited number of iterations) and discount factor equals to 1.0. In the comparisons we use the EU for the agents' initial state ($\{cell{=}1, light{=}green\}$).

Figure 4 shows the results. The vertical axis indicates the EU for the agent in the initial state, while the horizontal axis indicates the detection rate of norm violations in the RMAS. The NC Agent does not violate the $norm_1$, therefore her EU is constant (0.6500, dotted line) – it does not depend on the reactions, and consequently, does not depend on the detection rate. On the other hand, for the NNC Agent, the EU decreases as the detection rate increases until it reaches the EU equals to 0.6500. At this point, the

NNC Agent has assumed a norm-compliant behaviour because the EU of being non-normative are lower than the EU of being norm-compliant. In the worst case, the EU obtained by NNC Agent is equal to the EU obtained by the NC Agent. For example, assume the $reaction_2$ is executed when the $norm_1$ is violated. In this case, the NNC Agent breaks the $norm_1$ only if the detection rate is under 50%. Otherwise, she follows the $norm_1$ and executes *skip* when the traffic light is red.

Comparing the effect of the four different reactions under the *same detection rate*, we conclude the EU is lower for the $reaction_1$ (most restrictive) and higher for the $reaction_4$ (most relaxed). The only setting where the NNC Agent never follows the $norm_1$ is when the $reaction_4$ is the chosen one. In this case, the $reaction_4$ is too soft.

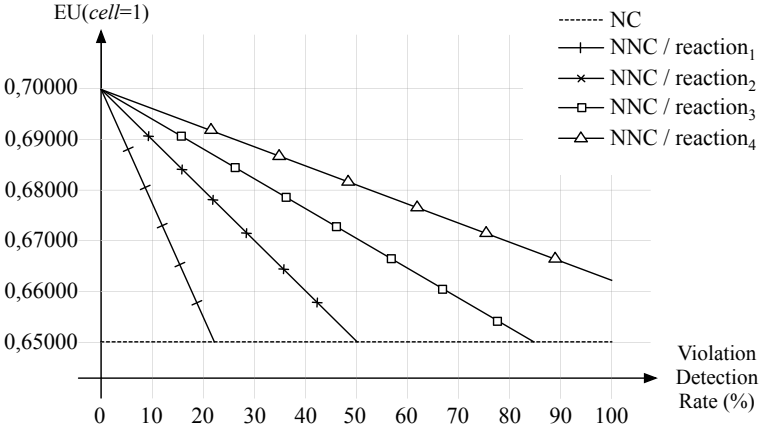


Fig. 4. Chart showing the agents' expected utility (EU) for the initial state, taking into account the detection rate of norm violations in the RMAS. Each line depicts the EU for a particular agent (NC or NNC) with different reactions for violating the norm governing the RMAS.

5 Related Work

The idea of norm violation by autonomous agents has been discussed by the MAS community in the past years. Most work on normative reasoning has been done from the practical reasoning perspective, specially through the well-known *goal-oriented* architectures [5, 7, 2, 3, 11, 13]. In these models, usually, there is some deliberation for weighting competing alternatives, which can be *personal goals* of the agent or *normative goals* generated from the norms ruling the MAS. If these normative goals survive the deliberation process, then the agent complies with the respective norms, otherwise norm violations may take place. Assuming the deliberation is done on the basis of a *preference ordering*, the main problem consists of computing the preference values to be assigned to the normative goals. However, the impact of the planning task (also referred to as *means-end reasoning*) on the normative reasoning has received less attention.

Another research direction on norm violations by autonomous agents consists of employing essentially *utility-oriented* agents. Cardoso and Oliveira [4] study adaptive

mechanisms that enable a normative framework to change deterrence sanctions according to an agent population. The agents are characterized by a risk tolerance parameter and a social awareness parameter. Although these agents are utility maximizers, they are not all equally self-interested, what makes possible the generation of heterogeneous populations. In this work, the agents decide to violate a norm exclusively based on those two parameters. The relation between norm deterrence and planning is not considered. Ågotnes *et. al.* [1] develop a model of normative multiagent system, whose possible transitions are represented through Kripke structures and norms are implemented as constraints over these structures. The agent's goals are specified as a hierarchy of formulae of Computational Tree Logic. Such hierarchy defines a model of ordinal utility, what makes possible the interpretation of the Kripke-based normative systems as games. This paper focuses on the problem concerning whether the agents should defect or not from the normative system. To address this question, the authors use game theory: an agent takes into account not just whether the normative system would be beneficial for itself, but also whether other agents will rationally choose to participate. The normative reasoning is based on the utility of the goals and the pay-offs for the game. It is assumed the reasoning process takes place before the agents join the system (*design time*).

6 Conclusion

This paper presents an architecture for adaptive norm-aware agents that estimates the benefits for breaking norms. Considering such estimation, the agent decides to comply or not with norms. We use MDPs for normative reasoning. It makes explicit the cost of planning with norms (amount of time to make a rational choice in a regulated environment) and the benefits of the plans (through the expected rewards). Internally, the agent does not assign preference values or desirability degrees to the normative structures. Instead, the impact of the norms and reactions on the agent is observed in her expected utilities and policy.

Our results have shown that the presented agent model obtains more rewards than its norm-compliant version. However, the main drawback of the model when compared with the norm-compliant agent is the amount of time taken to compute the policy. Our first future work concerns the exploration of techniques to improve the efficiency of the reasoning process. Among these possible techniques, we cite the Factored MDPs [10] and hybrid agent architectures [12]. Our second future work consists of experimenting our model in a RMAS with multiple agents, where the norms can dynamically change in order to fulfil successfully their purpose.

Acknowledgements

This work is supported by the Spanish Ministry of Science and Innovation through the projects "AT" (CSD2007-0022, CONSOLIDER-INGENIO 2010) and "OVAMAH" (TIN2009-13839-C03-02).

References

1. Ågotnes, T., van der Hoek, W., Wooldridge, M.: Normative system games. In: 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, pp. 1–8. IFAAMAS (2007)
2. Boella, G., Damiano, R.: An Architecture for Normative Reactive Agents. In: Kuwabara, K., Lee, J. (eds.) PRIMA 2002. LNCS (LNAI), vol. 2413, pp. 1–17. Springer, Heidelberg (2002)
3. Broersen, J., Dastani, M., Hulstijn, J., van der Torre, L.: Goal Generation in the BOID Architecture. Cognitive Science Quarterly Journal 2(3-4), 428–447 (2002)
4. Cardoso, H.L., Oliveira, E.: Adaptive Deterrence Sanctions in a Normative Framework. In: Proceedings of the 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2009), Milan, Italy, September 15-18, pp. 36–43. IEEE, Los Alamitos (2009)
5. Castelfranchi, C., Dignum, F., Jonker, C., Treur, J.: Deliberative Normative Agents: Principles and Architecture. In: Jennings, N.R. (ed.) ATAL 1999. LNCS, vol. 1757, pp. 364–378. Springer, Heidelberg (2000)
6. Centeno, R., Billhardt, H., Hermoso, R., Ossowski, S.: Organising MAS: A Formal Model Based on Organisational Mechanisms. In: Proceedings of the 2009 ACM Symposium on Applied Computing (SAC), Honolulu, Hawaii, USA, March 9-12, pp. 740–746. ACM, New York (2009)
7. Dignum, F., Morley, D.N., Sonenberg, L., Cavedon, L.: Towards socially sophisticated BDI agents. In: 4th International Conference on Multi-Agent Systems (ICMAS 2000), Boston, MA, USA, July 10-12, pp. 111–118. IEEE Computer Society, Los Alamitos (2000)
8. Fagundes, M.S., Billhardt, H., Ossowski, S.: Behavior Adaptation in RMAS: An Agent Architecture based on MDPs. In: 20th European Meeting on Cybernetics and Systems Research (EMCSR 2010), Vienna, Austria, April 6-9, pp. 453–458. Austrian Society for Cybernetic Studies (2010)
9. Grossi, D., Aldewereld, H., Dignum, F.: Ubi Lex, Ibi Poena: Designing Norm Enforcement in E-Institutions. In: Noriega, P., Vázquez-Salceda, J., Boella, G., Boissier, O., Dignum, V., Fornara, N., Matson, E. (eds.) COIN 2006. LNCS (LNAI), vol. 4386, pp. 101–114. Springer, Heidelberg (2006)
10. Guestrin, C., Koller, D., Parr, R., Venkataraman, S.: Efficient Solution Algorithms for Factored MDPs. Journal Artificial Intelligence Research (JAIR) 19, 399–468 (2003)
11. y López, F.L., Luck, M., d'Inverno, M.: A Normative Framework for Agent-Based Systems. In: Normative Multi-agent Systems. Dagstuhl Seminar Proceedings, vol. 07122 (2007)
12. Nair, R., Tambe, M.: Hybrid BDI-POMDP Framework for Multiagent Teaming. J. Artif. Intell. Res (JAIR) 23, 367–420 (2005)
13. Pacheco, N.C., Argente, E., Botti, V.: A BDI Architecture for Normative Decision Making (extended abstract). In: 9th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, pp. 1383–1384. IFAAMAS (May 2010)

Query-Based Argumentation in Agent Programming*

Sebastian Gottifredi, Alejandro J. Garcia, and Guillermo R. Simari

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Artificial Intelligence Research and Development Laboratory,
Department of Computer Science and Engineering - Universidad Nacional del Sur,
Bahía Blanca, Argentina
{sg,ajg,grs}@cs.uns.edu.ar

Abstract. In this work we will present an integration of a query-answering argumentation approach with an abstract agent programming language. Agents will argumentatively reason via special context-based queries, using their mental information. A set of mind affecting capabilities will allow the agent to dynamically modify its mental components and the argument comparison criterion. We will show the models for deliberation and execution of this integrated framework.

1 Introduction

In this work, we present a declarative agent programming language (APL) that uses argumentation as a reasoning formalism. Agent mental components of the proposed APL will be represented in Defeasible Logic Programing (DeLP) [7]. Hence, an agent developer will be able to represent conflicting goals and beliefs, and the argumentation formalism will be used for deciding which beliefs and goals are regarded as warranted. Our approach follows the spirit of 3APL [5], where interactions between mental components are made through queries in the sense of logic programming; therefore we will use DeLP contextual queries [6] to connect the mental components. Free variables in these queries will allow to belief and goal information in plans. In addition, agent capabilities are combined with the argumentative formalism, to dynamically modify the DeLP programs used to represent its mental components. We will present a set of semantic rules to show agent execution dynamics, and properties of our proposed APL.

Recently, logic based APLs have received special attention [5,3]. These languages provide interesting features like declarative specifications, clear reasoning semantics, and verifiable development. However, none of them use an argumentative formalism for agent reasoning. Since the aim of this research is related to agent specification and implementation, our approach will be inspired on these languages.

Several articles in the literature recognize the relevance of using argumentation as the reasoning mechanism of agent systems [2]. Since the beginnings of BDI [4], the use of defeasible reasoning to handle intentional reasoning was considered. Nowadays, [1,11,10,8] present important contributions showing how BDI agent mental components can be benefited with the use of argumentation for reasoning with conflicting information. However, in the field of APL, although deliberation is an important topic,

* Partially supported by UNS (PGI 24/ZN18) and CONICET (PIP 112-200801-02798).

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 284–295, 2010.

it is necessary to establish execution semantics (or agent state dynamics). That is, in APLs area is important to specify rules that show how actions will affect agent mental components, and thus agent future reasoning. On the one hand [1,11,10] propose BDI architectures (instead of APLs), therefore, their main focus resides in the argumentative support for agent deliberation and not in agent execution dynamics. On the other hand [8] presents an APL with an execution model for their agents, but the mental components can not be dynamically modified by the agent, thus the argumentation formalism is not affected by agent dynamics. The APL presented in this work aims to bridge these gaps, integrating argumentation into agents execution and deliberation models.

2 DeLP-Server Basis

In order to provide an argumentative reasoning service for multi-agent systems, an implementation of DeLP, called DeLP-server, has been developed [6]. A DeLP-server is a stand-alone application that stores a DeLP-program used to answer client queries.

In DeLP [7], knowledge is represented using facts, strict rules and defeasible rules. *Facts* are ground literals representing atomic strict information or its negation using the strong negation "$\sim$". *Defeasible Rules* (d-rules) are denoted $L_0 \prec L_1, \ldots, L_n$ (where L_i are literals) and represents defeasible information. This kind of information may be used if nothing could be posed against it. A d-rule *"*$Head \prec Body$*"* expresses that *"reasons to believe in Body give reasons to believe in Head"*. D-rules can contain free variables and they will be instantiated by DeLP with the usual substitution conventions. In this paper we will consider a restricted form of program that do not have strict rules.

Definition 1 (Restricted Defeasible Logic Program). *A restricted defeasible logic program (de.l.p. for short) is a triplet* $\mathcal{P} = (\Psi, \Delta, \prec)$*, where* Ψ *is a set of facts,* Δ *is a set of d-rules, and* $\prec$ *is an argument comparison criterion.*

Strong negation is allowed in the head of program rules, and hence, may be used to represent contradictory knowledge. Given a literal L the complement with respect to strong negation will be noted $\overline{L}$. From $(\Psi, \Delta, \prec)$ contradictory literals could be derived, however, Ψ must possess certain internal coherence (it has to be non-contradictory).

To deal with contradictory information, in DeLP, *arguments* for conflicting pieces of information are built and then compared to decide which one prevails. The prevailing argument is a *warrant* for the information that it supports. In DeLP, a query L is *warranted* from a program $(\Psi, \Delta, \prec)$ if a *non-defeated* argument $\mathcal{A}$ supporting L exists.

An *argument* $\mathcal{A}$ for a literal L, denoted $\langle \mathcal{A}, L \rangle$($\mathcal{A}$ for short), is a minimal set $\mathcal{A}$ of ground instances of d-rules from Δ, such that $\mathcal{A} \cup \Psi$ is non-contradictory and there is a derivation for L from $\mathcal{A} \cup \Psi$. To establish if $\mathcal{A}$ is a non-defeated argument, *defeaters* for $\mathcal{A}$ are considered, *i.e.*, counter-arguments that are preferred over $\mathcal{A}$. In DeLP this preference is determined by the argument comparison criterion (see Def. 1), which defines a partial order among arguments. In particular, DeLP provides a modular implementation of the comparison criterion, thus, the most appropriate criterion can be selected. In the examples we will use *generalized specificity* [7], usually noted $\prec_{GE}$, a criterion that favors two aspects of an argument: it prefers (1) a more precise argument (*i. e.*with greater information content) or (2) a more concise argument (*i. e.*with less use of rules).

A defeater $\mathcal{D}$ for an argument $\mathcal{A}$ can be *proper* ($\mathcal{D}$ is preferred to $\mathcal{A}$) or *blocking* (same strength). Since defeaters are arguments, there may exist defeaters for them, defeaters for these defeaters, and so on. Thus, leading to an exhaustive tree structure called *dialectical tree*, where, every node (except the root) is a defeater of its parent, and leaves are non-defeated arguments (for more details see [7]). In a dialectical tree every node can be marked as *defeated* or *undefeated*: leaves are marked as undefeated nodes, and inner nodes are marked defeated when there is at least a child marked undefeated, or are marked undefeated when all its children are marked defeated.

Dialectical trees are used to determine if a ground queried literal (DeLP-Query) is warranted or not form a *de.l.p.* Here we allow free variables in the queried literals, allowing agents to pass belief and goal resulting substitutions to other components.

Definition 2 (Warrant). *Let $\mathcal{P}$ be a de.l.p. and Q a DeLP-Query, then Q is warranted from $\mathcal{P}$ under the substitution τ (noted $\mathcal{P}|\sim_w Q\tau$[1]) iff there exists an argument $\langle \mathcal{A}, Q\tau \rangle$, such that $\mathcal{A}$ is un undefeated root of a dialectical tree and τ is the substitution used to bound the free variables of Q. If Q is not warranted from $\mathcal{P}$, it will be noted $\mathcal{P} \not|\sim_w Q$.*

Proposition 1. *Given a de.l.p. $\mathcal{P}$, then the set $W = \{Q_i \,|\mathcal{P}|\sim_w Q_i\}$ is non-contradictory.*[2]

To answer queries, a DeLP-Server will use the public knowledge stored in it, together with individual knowledge that clients can send attached to a query, creating a particular *context* for that query (see [6] for details). This context is knowledge that the server will use for answering the query and will not affect other future queries. The temporal scope of the context sent in a contextual query is limited and disappears once the query has been answered. Since contextual information can be in contradiction with the information stored in the server, different types of contextual queries were defined in [6]. Next, we will present a contextual query specifically designed for this work, which is used to model the interaction of the mental components of the proposed APL.

Definition 3 (Contextual query). *Let $\mathcal{P} = (\Psi, \Delta, \prec)$ be a de.l.p., a contextual query for $\mathcal{P}$ is a pair $[(\Phi^+, \Theta^-), Q]$ where Q is a DeLP-query, $\Phi^+$and $\Theta^-$are non-contradictory sets of literals.*

This special kind of contextual query will temporarily add and remove elements from $\mathcal{P}$. The literals of Φ^+ will be added as facts to Ψ in a way that its consistency is not harmed. The literals of Θ^- will determine two actions: the literals to be removed from Ψ and the rules to be removed from Δ (whose head matches them).

Definition 4 (Contextual query warrant). *Let $\mathcal{P}$= (Ψ,Δ,$\prec$) be a de.l.p., the contextual query $[(\Phi^+, \Theta^-), Q]$ is warranted from $\mathcal{P}$ under substitution τ (noted $\mathcal{P}|\sim_w [(\Phi^+, \Theta^-), Q\tau]$) iff $\mathcal{P}_{cq}|\sim_w Q\tau$ where $\mathcal{P}_{cq} = (((\Psi \oplus \Phi^+) \setminus \Theta^-, (\Delta \ominus \Theta^-)), \prec)$, with $(\Psi \oplus \Phi^+) = (\Psi \setminus \{\overline{L} | L \in \Phi^+\}) \cup \Phi^+$, and $(\Delta \ominus \Theta^-) = \{r_i | r_i = L_h \prec Body \in \Delta \wedge L_h \notin \Theta^-\}$.*

Proposition 2. *Let $[(\Phi^+, \Theta^-), Q]$ be a contextual query for a de.l.p. $\mathcal{P}$= (Ψ,Δ,$\prec$). Then the program $\mathcal{P}_{cq}$ used to determine the status of Q, is a valid de.l.p.*

[1] In some contexts we will omit the substitution τ from the notation of warrant.

[2] Some proofs were omitted due to space reasons.

Proof: By Def. 4 $\mathcal{P}_{cq} = (((\Psi \oplus \Phi^+) \setminus \Theta^-, (\Delta \ominus \Theta^-)), \prec)$ will be used to determine the status of Q. Ψ is consistent because $\mathcal{P}$ is a *de.l.p.*, Φ^+ is consistent by Def. 3, and by Def. 4 $(\Psi \oplus \Phi^+)$ is consistent. Removing literals or d-rules does not harm consistency, then $\mathcal{P}_{cq}$ is a *de.l.p.* □

Proposition 3. *Given a de.l.p. $\mathcal{P}$, if $\mathcal{P} \sim_w [(\Phi^+, \Theta^-), Q]$ then $\mathcal{P} \not\sim_w [(\Phi^+, \Theta^-), \overline{Q}]$.*

Proof: By Prop. 2, the output $\mathcal{P}_{cq}$ of applying Φ^+, Θ^- to $\mathcal{P}$ is a *de.l.p.* By Prop. 1 the set of warranted queries W from a *de.l.p.* $\mathcal{P}_{cq}$ is non-contradictory, and by hyp. $Q \in W$, then $\overline{Q} \notin W$. Finally, $\mathcal{P} \not\sim_w [(\Phi^+, \Theta^-), \overline{Q}]$ □

Example 1. *Suppose an agent representing a free-lance business man that travels to different cities to work. This agent will get possible businesses from external sources (newspaper, internet, contacts, etc.) and use the de.l.p. $\mathcal{PB}_1 = (\Psi_1^B, \Delta_1^B, \prec_{GE})$ to decide if these businesses are good and profitable opportunities, with:*

$$\Delta_1^B = \left\{ \begin{array}{l} opp(X,C) \prec biz(X,C,P), \\ \sim opp(X,C) \prec biz(X,C,P), risky(C), \\ risky(C) \prec closing(C), \\ risky(C) \prec fusion(C,D), \\ \sim risky(C) \prec fusion(C,D), strong(D) \end{array} \right\} \quad \Psi_1^B = \left\{ \begin{array}{ll} fusion(a,b), & far(p1,p2), \\ strong(b), & far(p2,p3), \\ closing(c), & near(p1,p3), \\ money(100), & at(p1), \end{array} \right\}$$

In Δ_1^B, for instance, the first rule denotes that there is a good business opportunity in the city X with the company C, if there is a possible business in that city X with company C having a tentative profit of P; and the last one denotes that a company C is not risky if it is in fusion with a strong company D. The set Ψ_1^B contains information about the companies, the city where the agent is, the distances between the cities and agent's money. Note that there is no information about biz(X,C,M) in Ψ_1^B, this is because possible businesses are obtained externally. The agent will use contextual queries to add business information to $\mathcal{PB}_1$, and determine if there is a warranted opportunity. For example, the contextual query $[(\{biz(p1,a,3), biz(p1,c,5)\}, \emptyset), opp(p1,a)]$ is warranted, while $[(\{biz(p1,a,3), biz(p1,c,5)\}, \emptyset), opp(p1,c)]$ is not.

3 APL Formal Concepts

In this section, we will show how the defeasible argumentation can be integrated into APLs using DeLP-Servers. We will introduce a declarative APL, called DeLPAP. This language is partially inspired in 3APL [5], where interactions among mental components are made through queries in the sense of logic programing. Following that model we will use contextual queries to model the interaction between agent components.

3.1 Syntax

Agents in DeLPAP are characterised by three components: belief and goal bases, and a set of plan rules. These bases represent agent mental components, and will be specified by *de.l.p.*s, which are used to compute agents beliefs and goals in each deliberative cycle. The plan rules will be used to generate plans in order to achieve goals, using actions to interact with the environment and modify its mental components.

Beliefs are used to represent the information that the agent has and infers about the world. In DeLPAP, the belief base will be represented by a *de.l.p.* denoted $\mathcal{PB} = (\Psi^B, \Delta^B, \prec^B)$. Therefore, a developer can specify potentially contradictory information using the d-rules. For instance, the *de.l.p.* $\mathcal{PB}_1$ of Ex. 1 can be a belief base of the businessman agent used to determine if a business is a good profitable opportunity.

Perceptual information will be also considered as beliefs, and therefore, it may appear in the body of belief base d-rules. For instance, in Ex. 1, literals *biz(X,C,M)* are considered as a perception and used in some d-rules. All perceptual elements will be identified by the set of literals E. However, we will explain how the $\mathcal{PB}$ interacts with E in the semantics subsection, since E will dynamically change as the world evolves.

In order to refer to Beliefs from the other components, we will use special literals $B(L)$ (with L a literal) called *current beliefs*. $B(L)$ denotes that the agent believes in *L*. Later we will explain how these special literals are calculated from the belief base.

Goals are used to denote situations of the world that the agent want to realize. The goal base has knowledge about agent goals, and it is used to determine which of these goals are feasible for the agent in each deliberative cycle. In agent programming, goals may be unconditional or conditional. Unconditional (or independent) goals are always adopted. Conditional goals are not always adopted, they depend on other goals and beliefs. Goals may also be conflicting, *i. e.*, some goals may not be adoptable at the same time. In DeLPAP goals will be represented by a *de.l.p.*, where: unconditional goals will be modeled using facts, and conditional goals using d-rules. Conflicting goals can be represented using strong negation. The formalism will identify the goals that are in conflict, and the argumentation process will be used for deciding which one to adopt. Current beliefs in the body of d-rules will be used to obtain belief information.

Definition 5 (Goal Base). *The goal base is a de.l.p.* $\mathcal{PG} = (\Psi^G, \Delta^G, \prec^G)$*, where* Δ^G *has rules of the form* $(L_h \prec B(L_0), \ldots, B(L_k), L_{k+1}, \ldots, L_n)_{k+n\geq 0}$ *and* L_i *a Literal.*

Example 2. *Continues Ex. 1, where* $\mathcal{PB}_1$ *is the agent belief base, consider the goal base* $\mathcal{PG}_1 = (\emptyset, \Delta_1^G, \prec_{GE})$ *used by the agent to determine if it wants to be at some city (to do business) or to work in the city that it currently is at.*

$$\Delta_1^G = \left\{ \begin{array}{l} at(X) \prec B(opp(X,C)), \\ \sim at(X) \prec at(Y), B(biz(X,\_,P)), B(biz(Y,\_,N)), P < N, \\ work(X,C) \prec B(at(X)), B(opp(X,C)), \\ \sim at(X) \prec work(X,C) \end{array} \right\}$$

For example, the first rule expresses that if it believes that there is an opportunity in city X, it has reasons for wanting to be at X (the agent has reasons for wanting to reach a state of the world where be at(X) is an actual belief); and the second rule denotes that if it wants to be at city Y and in Y it can do a better business than at city X, there is a reason against wanting to be at X.

In order to refer to Goals from the other components, we will use special literals $G(L)$ (with *L* a literal) called *current goals*. $G(L)$ denotes that the agent wants to achieve L. Later we will explain how to obtain these literals from the goal base.

Basic actions specify capabilities of an agent, i.e., actions that an agent can perform as part of a plan to achieve a desirable situation. Below we will present three types of basic actions: mental actions, criterion change actions, and external actions.

A **mental action** updates the belief base or the goal base when executed. This kind of action can be used to store information received from other agents or environments, to remove unwanted goals, change the agents behavior, or to store temporarily data or the results of some computations. Such an action is specified in terms of pre-,post-conditions and the target mental component.

Definition 6 (Mental Action). *A Mental Action (or ma for short) is a tuple* $Ma = (Base, \beta, Name, Pos)$*, where Base is a de.l.p. representing the mental component that will be affected by the action,* $\beta = \{B(L_0), \ldots, B(L_k)\}_{k \geq 0}$*, Name is an atom used to call the Ma from a plan, and* $Pos = [-Y_1, \ldots, -Y_m, X_1, \ldots, X_n]_{n \geq 0, m \geq 0}$*, where* X_i *is an element to be added to Base and* Y_j *is an element to be removed from Base.* X_i *and* Y_j *can be a literal or a d-rule. The set of all ma available for agent is noted* S_{ma}*.*

Example 3. *Consider the following set of mental actions that refers to the* $\mathcal{PB}_1$ *of Ex.1*

$$S_{ma1} = \left\{ \begin{array}{l} (\mathcal{PB}, \{B(money(M)), M > S\},\ spend(S),\ [-money(M), money(M-S)]) \\ (\mathcal{PB}, \{B(money(M))\},\ earn(E),\ [-money(M), money(M+E)]) \\ (\mathcal{PB}, \{B(at(Y))\},\ travel(X),\ [-at(Y), at(X)]) \end{array} \right\}$$

for instance, the first ma denotes that to spend an amount of money S, the agent should remove the old amount of money (M) and add the new one (M-S), if it believes that M is greater than S.

A **criterion change action** changes (completely) the comparison criterion of the belief base or goal base when executed. Like mental actions, criterion change actions are specified in terms of pre-,post-conditions and the target mental component.

Definition 7 (Criterion Change Actions). *A Criterion Change Action (cca for short) is a triplet* $CCa = (Base, \beta, Name, CC)$*, where Base is a de.l.p. representing the mental component that will be affected by the action,* $\beta = \{B(L_0), \ldots, B(L_k)\}_{k \geq 0}$*, Name is an atom used to call the CCa from a plan, and CC the name of a comparison criterion. The set of all cca is noted* S_{cca}*.*

An **external action** is used to interact with the environment or for communication purposes. The effects of external actions are assumed to be determined by the environment and might not be known by the agent beforehand. We will treat these actions abstractly, and specify them as simple atoms.

Plans are used to achieve goals. In this article we will assume that a plan will be simple sequences of actions of the form $[a_1, \ldots, a_m]$. More complex plan structures like the ones presented in 3APL can be easily adapted to `DeLPAP`. However, to simplify the explanation of the `DeLPAP`, we will leave the plan structure as elementary as possible.

In order to decide how to act, `DeLPAP` agents will use **plan rules**, which are based in the reasoning rules used in 3APL. These rules will establish a mapping between goals and plans. Basically, a plan rule will determine the plan to execute with the objective of achieving its associated goal, when some belief preconditions are met.

Definition 8 (Plan Rule). *A plan rule has the form* $\rho = \kappa \leftarrow \beta \mid \pi$*, where* $\kappa = G(K)$*,* $\beta = \{B(P_1), \ldots, B(P_n)\}$ *with* $n \geq 0$ *and* π *a plan. The set of plan rules is denoted* $\mathcal{R}$*.*

Example 4. *Continuing our example, consider that the agent has the plan rule set:*

$$\mathcal{R}_1 = \left\{ \begin{array}{l} G(at(X)) \leftarrow B(at(Y)), B(near(X,Y)) \mid [train(X,Y,S), spend(S), travel(X)] \\ G(at(X)) \leftarrow B(at(Y)), B(far(X,Y)) \mid [plane(X,Y,S), spend(S), travel(X)] \\ G(work(X,C)) \leftarrow B(at(X)) \mid [doBiz(X,C,M), earn(M)] \end{array} \right\}$$

For instance, the first plan rule expresses that if the agent wants to be at city X, and believes that the city where it is (Y) is near X, then it should first obtain the cost S of travelling from Y to X by train using an external action, then spend the travelling amount S, and finally travel to X.

As in most declarative APLs, to program a DeLPAP agent means to specify the *de.l.p.*s of the mental components, the basic actions and the plan rule set. For instance, a DeLPAP agent can be specified using the belief base of Ex.1, the goal base of Ex.2, the mental actions of Ex.3 and the plan rule set of Ex. 4.

3.2 Semantics

In this section, we will define DeLPAP formal semantics, which will introduce the processes to handle practical reasoning and execution dynamics. Therefore here we will describe how to determine warranted goals and beliefs, how to use these elements to establish which plans are executable, and how to finally execute a plan.

Following the style from the semantics specification of other APLs such as AgentSpeak(L) and 3APL, the semantics will be given using a transition system. A transition system is a set of derivation rules for deriving transitions. A transition corresponds to a single computation step, it transforms a configuration C into another C' iff some condition holds. These configurations will be used to denote the agent states, thus, transition rules will show how an agent moves from one state to another.

A configuration of a DeLPAP agent will show a snapshot of its current mental components at a certain moment. Configurations contain only information about the components that can be modified during the lifetime of the agent (beliefs, goals, plans and perceptions). Together with a substitution component, these changing components constitute a DeLPAP agent configuration. The substitution part of the configuration is used to store values or bindings associated with variables.

Definition 9 (Configuration). *A configuration of a* DeLPAP *agent is a tuple* $\langle E, \mathcal{PB}, \mathcal{PG}, \Pi, \theta \rangle$ *where* E *is a set of literals denoting agent perceptions;* $\mathcal{PB}$ *is a belief base,* $\mathcal{PG}$ *is a goal base;* Π*, the plan base, is a pair* (π, κ)*, denoting the currently adopted plan* π*, and* κ *the currently adopted goal to be achieved with* π*; and* θ *is a ground substitution that binds domain variables to ground terms.*

We will use ϵ (the empty plan base) to denote that the agent has no adopted plan and no adopted goal. The initial conf. contains the initial goal and belief bases $\mathcal{PG}_0, \mathcal{PB}_0$, an empty plan base, and an empty substitution component, that is $\langle E, \mathcal{PB}_0, \mathcal{PG}_0, \epsilon, \emptyset \rangle$.

From a given conf. it is possible to determine the agent's *current beliefs*, which will be literals warranted from the belief base. In current beliefs warranting process, perceptions should be taken into account. Thus, we will use the contextual queries (Def. 3) to add (and revise if necessary) the perceptual information to the belief base.

Definition 10 (current belief). *Let $C = \langle E, \mathcal{PB}, \mathcal{PG}, \Pi, \theta \rangle$ be a conf., a literal L will be a current belief in the conf. C under the substitution τ (noted $C \vdash_B B(L)\tau$) iff $\mathcal{PB}|\sim_w[(E, \emptyset), L\tau]$. We will also say that a set of literals $S = \{L_0, \ldots, L_n\}_{n \geq 0}$ is a set of current beliefs in C under τ (noted $C \vdash_B \{B(L_0), \ldots, B(L_n)\}\tau$) iff $\forall L_i \in S$ $C \vdash_B B(L_i)\tau$. The set of all current beliefs in the configuration C will be denoted $\boldsymbol{B}_C$.*

Proposition 4. *Let $C = \langle E, \mathcal{PB}, \mathcal{PG}, \Pi, \theta \rangle$ be a conf., if $L \in E$ then $C \vdash_B B(L)$.*

Proof: By Def.4, $L \in (\Psi^B \oplus E)$, therefore, there will be an undefeated argument for L in $(((\Psi^B \oplus E) \setminus \emptyset, (\Delta^B \ominus \emptyset))$. Then, $\mathcal{PB}|\sim_w[(E, \emptyset), L]$, and $C \vdash_B B(L)$ □

Proposition 5. *Let $C = \langle E, \mathcal{PB}, \mathcal{PG}, \Pi, \theta \rangle$ be a conf., if $C \vdash_B B(L)$ then $C \nvdash_B B(\overline{L})$.*

Proof: Straight forward from Prop.3 and Def. 10.

Example 5. *Suppose an agent conf. $C_1 = \langle E, \mathcal{PB}_1, \mathcal{PG}_1, \epsilon, \emptyset \rangle$, with $\mathcal{PB}_1$ and $\mathcal{PG}_1$ of examples. 1 and 2 respectively, and $E = \{$biz(p1,c,20), biz(p2,a,30), biz(p3,b,100)$\}$. Then in C_1 current beliefs will be all the elements in E, Ψ_1^B and $\{$ B(opp(p2,a)), B(opp(p3,b)), B(∼opp(p1,c)), B(∼risky(a)), B(risky(c))$\}$. Note that $C_1 \nvdash_B B(opp(p1, c))$ because opp(p1,c) supporting argument is defeated by the argument for ∼opp(p1,c), which is supported by c being a risky company.*

Goal semantics describe the goals that an agent want to achieve in a given configuration. These goals are called *current goals*, and will be the warranted literals of the goal base. However, using only the information of the goal base is not enough to determine these goals: goal arguments may involve beliefs, and also already achieved goals should not be considered as current goals. Therefore, current beliefs are needed. We will capture these notions using the previously presented contextual queries.

Definition 11 (current goal). *Let $C = \langle E, \mathcal{PB}, \mathcal{PG}, \Pi, \theta \rangle$ be an agent configuration, a literal L will be a current goal in the configuration C under the substitution τ (noted $C \vdash_G G(L)\tau$) iff $\mathcal{PG}|\sim_w[(\boldsymbol{B}_C, A), L\tau]$, where $A = \{L_i | B(L_i) \in \boldsymbol{B}_C\}$.*

Proposition 6. *Let $C = \langle E, \mathcal{PB}, \mathcal{PG}, \Pi, \theta \rangle$ be a conf., if $C \vdash_G G(L)$ then $C \nvdash_G G(\overline{L})$.*

Proof: Straight forward from Prop.3 and Def. 11.

Example 6. *Consider the conf. C_1 of Ex. 5, in C_1 the current goal will be G(at(p3)). Note that $C_1 \nvdash_G G(at(p2))$ since at(p2) argument is defeated by an argument establishing that the agent can not be in two places at the same time, and that in p3 there is a more profitable opportunity.*

As introduced above, the semantics of current goals should only consider those unachieved goals. Recall that goals represent situations of the world that the agent wants to realise. Therefore, if an agent believes in a literal L, the goal for L should be considered as achieved and then ignored in the current goals warranting process.

Proposition 7. *Let $C = \langle E, \mathcal{PB}, \mathcal{PG}, \Pi, \theta \rangle$ be a conf., if $C \vdash_B B(L)$ then $C \nvdash_G G(L)$. Proof:* By Def.10 $B(L) \in \mathbf{B}_C$. Let $A = \{L_i | B(L_i) \in \mathbf{B}_C\}$, then $L \in A$. From the *de.l.p.* $\mathcal{PG}_{cq} = (((\Psi^G \oplus \mathbf{B}_C) \setminus A, (\Delta^G \ominus A))$ there is no argument for L,

since by Def.4 there will be no fact L in $(\Psi^G \oplus \mathbf{B}_C) \setminus A$ or d-rules with head L in $\Delta^G \ominus A$. Therefore $\mathcal{PG}_{cq} \not\sim_w L, \mathcal{PG} \not\sim_w [(\mathbf{B}_C, A), L]$, and $C \not\vdash_G G(L)$ □

Example 7. *Suppose that the agent of Ex. 5 is in a new situation, it has achieved the goal of being at(p3). This situation is described by the conf.* $C_2 = \langle E, \mathcal{PB}_2, \mathcal{PG}_1, \epsilon, \emptyset \rangle$, *where* $\mathcal{PB}_2 = (\Psi_1^B \setminus \{at(p1)\} \cup \{at(p3)\}, \Delta_1^B, \prec_{GE})$ *and* $E, \Delta_1^B, \mathcal{PG}_1$ *are the same as in Ex.5. Therefore from* C_2 *the current goal will be G(work(p3,b)). Note that, now* $C_2 \vdash_B B(at(p3))$, *thus, at(p3) was not considered in current goals warranting process.*

Using the current beliefs and goals, the agent will be able to determine which of its plan rules are applicable. Basically, a plan rule will be applicable iff its associated goals are current goals and its associated beliefs are current beliefs. Once an applicable plan rule is found, the plan and the goal of the rule are adopted. However, plan rules will be only applied when there is no adopted plan.

Definition 12 (Plan Selection). *Let* $C = \langle E, \mathcal{PB}, \mathcal{PG}, \epsilon, \emptyset \rangle$ *be an agent configuration,* $\mathcal{R}$ *the agent plan rules set,* $\rho = \kappa \leftarrow \beta \mid \pi \in \mathcal{R}$, *and* τ_1, τ_2 *be ground substitutions.*

$$\frac{(C \vdash_G \kappa\tau_1) \ \wedge \ (C \vdash_B \beta\tau_1\tau_2)}{\langle E, \mathcal{PB}, \mathcal{PG}, \epsilon, \emptyset \rangle \rightarrow \langle E, \mathcal{PB}, \mathcal{PG}, (\pi, \kappa), \tau_1\tau_2 \rangle}$$

Note that τ_1 is the substitution returned by DeLP that should be used in order to bind the free variables of κ. After that, the substitution τ_1 is applied to β and DeLP returns the substitution τ_2 which binds free variables of $\beta\tau_1$. In addition, observe that $\tau_1\tau_2$ is established as the ground substitution of the new configuration.

Example 8. *Consider the conf.* C_1 *of Ex. 5 and the set* $\mathcal{R}_1$, *Since* $C_1 \vdash_G G(at(p3))$, $C_1 \vdash_B B(at(p1))$, *and* $C_1 \vdash_B B(near(p1, p3))$ *the first rule of* $\mathcal{R}_1$ *will be applicable. Then the agent plan base will be (*$[train(X, Y, S), spend(S), travel(X)]$, *G(at(p3))). Note that, in the resulting configuration the substitution component will contain the elements X/p3 and Y/p1.*

When a plan from a plan rule ρ is adopted, the agent commits to the goal of ρ and adopts its plan. This adopted plan will remain in execution until it is finished or when the adopted goal is no longer a current goal. Also as we will show below, the failure of an action preconditions removes the plan.

Next we will introduce the execution semantics of every action presented in the syntax subsection. Mental actions introduce permanent changes in the mental components. The following definition presents a function describing their effects.

Definition 13 (Update Function). *Let* $\mathcal{P}$ = (Ψ, Δ, $\prec$) *be a de.l.p., and* $nma\tau$ *the name of a mental action such that* τ *be a ground substitution and* $\exists$ *a mental action ma = (*$\mathcal{P}$, β, *nma, Pos), with* $Pos = [-Y_1, \ldots, -Y_m, X_1, \ldots, X_n]$. *Using the elements in Pos we define* $AddS = \{X_i\tau \mid \forall X_i$ *literal* $\}$, $DelS = \{\overline{X_i}\tau \mid \forall X_i$ *literal* $\} \cup \{Y_j\tau \mid \forall Y_j$ *literal* $\}$, *AddD=* $\{X_i\tau \mid \forall X_i$ *d-rule* $\}$, *DelD =* $\{Y_j\tau \mid \forall Y_j$ *d-rule* $\}$. *Then, the mental action update function is:* $\mathcal{T}(nma\tau, \mathcal{P}) = (((\Psi \setminus DelS) \cup AddS), ((\Delta \setminus DelD) \cup AddD), \prec)$

During the plan execution, if the mental action preconditions are met, the update function will be applied. In contrast, if the preconditions are not met or the adopted goal is no longer justified, the plan should be removed from execution.

Definition 14 (Mental Action Execution). *Let* $C = \langle E, \mathcal{PB}, \mathcal{PG}, ([nma, a_0, \ldots, a_m], \kappa), \theta\rangle$ *be conf., where nma the name of a mental action* $(\mathcal{PB}, \beta, nma, Pos)$*, and* τ *a ground substitution. The execution of ma is defined in two cases:*

$$\frac{(C \vdash_B \beta\theta\tau) \ \wedge \ (C \vdash_G \kappa) \ \wedge \ (\mathcal{T}(nma\theta\tau, \mathcal{PB}) = \mathcal{PB}')}{\langle E, \mathcal{PB}, \mathcal{PG}, ([nma, a_0, \ldots, a_m], \kappa), \theta\rangle \rightarrow \langle E, \mathcal{PB}', \mathcal{PG}, ([a_0, \ldots, a_m], \kappa), \theta\rangle}$$

$$\frac{C \nvdash_B \beta\theta \ \vee \ C \nvdash_G \kappa}{\langle E, \mathcal{PB}, \mathcal{PG}, ([nma, a_0, \ldots, a_m], \kappa), \theta\rangle \rightarrow \langle E, \mathcal{PB}, \mathcal{PG}, \epsilon, \emptyset\rangle}$$

Note that τ is not added to θ in the new conf. This is because τ is just used by the update function (see Def.13), and should not influence other actions in the adopted plan.

Example 9. *Consider* $C_3 = \langle E, \mathcal{PB}_1, \mathcal{PG}_1, ([spend(S), travel(X)], G(at(p3))), \theta\rangle$ *a conf. denoting that the agent is committed to be at(p3) and has already executed the first action of the plan adding* $S/8$ *(the train ticket costs 8) to* θ*. The ma* $spend(S)$ *(presented in Ex. 3) is executable since* $C_3 \vdash_B B(money(100))$ *and* $100 > 8$*. It produces* $\mathcal{PB}_3 = (\Psi_1^B \setminus \{money(100)\} \cup \{money(92)\}, \Delta_1^B, \prec_{GE})$ *leading to the conf.* $\langle E, \mathcal{PB}_3, \mathcal{PG}_1, ([travel(X)], G(at(p3))), \theta\rangle$

Transition rules for ma's to modify the goal base can be analogously defined. Next, we will show that in the DeLPAP strict knowledge always remains non-contradictory.

Proposition 8. *Given a* DeLPAP *agent conf.* $C = \langle E, (\Psi^B, \Delta^B, \prec^B), (\Psi^G, \Delta^G, \prec^G), \Pi, \theta\rangle$*, after any transition the sets* Ψ^B *and* Ψ^G *will remain non-contradictory.*

Proof: Straight forward from Def. 13, and Def. 14.

This and previous propositions show that although the knowledge representation language allows strong negation, if an agent starts with its Ψ sets consistent, regardless the action the agent executes or the changes in the environment, the inferences of the agent will be non-contradictory. Thus, with these properties we have shown that DeLPAP agents can use strong negation and be safe from inconsistencies.

Similarly to *ma* we will define the semantics of *cca*, which will change the argumentation comparison criterion of a mental component iff their preconditions are warranted.

Definition 15 (Criterion Change Actions execution). *Let* $C = \langle E, (\Psi^B, \Delta^B, \prec^B), \mathcal{PG}, ([ncca, a_0, \ldots, a_m], \kappa), \theta\rangle$ *be a conf.,* τ *a ground substitution, and ncca the name of a criterion change action* $(\mathcal{PB}, \beta, ncca, <)$*, with* $<$ *is an argument comparison criterion.*

$$\frac{(C \vdash_B \beta\theta\tau) \ \wedge \ (C \vdash_G \kappa)}{\langle E, (\Psi^B, \Delta^B, \prec^B), \mathcal{PG}, ([ncca, \ldots], \kappa), \theta\rangle \rightarrow \langle E, (\Psi^B, \Delta^B, <), \mathcal{PG}, ([a_0, \ldots], \kappa), \theta\rangle}$$

$$\frac{C \nvdash_B \beta\theta \ \vee \ C \nvdash_G \kappa}{\langle E, \mathcal{PB}, \mathcal{PG}, ([ncca, a_0, \ldots, a_m], \kappa), \theta\rangle \rightarrow \langle E, \mathcal{PB}, \mathcal{PG}, \epsilon, \emptyset\rangle}$$

The criterion change actions allow the agent to change the argument comparison criterion during plan's execution. Thus, for instance, the agent may have plans to changes this criterion to another which adapts to its current environment conditions, or receive a comparison criterion from another agent and use it. Finally, we define a transition rule to empty the adopted plan when it has been completely executed.

Definition 16 (Plan Finalization). *Let* $C = \langle E, \mathcal{PB}, \mathcal{PG}, ([\,], \kappa), \theta \rangle$ *be an agent conf.*

$$\frac{\top}{\langle E, \mathcal{PB}, \mathcal{PG}, ([\,], \kappa), \theta \rangle \rightarrow \langle E, \mathcal{PB}, \mathcal{PG}, \epsilon, \emptyset \rangle}$$

Now we can present a proposition which shows that when the adopted goal is achieved, its associated plan is removed from execution.

Proposition 9. *Let* $C = \langle E, \mathcal{PB}, \mathcal{PG}, (\pi, G(L)), \theta \rangle$ *be a conf., if* $C \vdash_B B(L)$ *then any possible transition will lead to the configuration* $C' = \langle E, \mathcal{PB}, \mathcal{PG}, \epsilon, \emptyset \rangle$

4 Conclusions and Related Work

In this work we presented an APL where mental components are represented and use DeLP as the reasoning mechanism. For the proposed APL we have shown how deliberation is made using the argumentation mechanism, and how this deliberation is affected by actions in operational terms. In particular, we used DeLP-Server contextual queries to handle mental component interaction. We have shown that contextual queries were really handy in that task because they modularize the interaction: they provide all the perceptual, belief and goal information that a component needs to answer the query, and the query itself, in one step. In addition the special contextual query that we presented allowed to represent the correct semantics for goal queries. That is, an agent will only pursue unachieved goals. We have shown how substitutions for free variables obtained by the DeLP argumentative mechanism can be used in a `DeLPAP` agent plan. We have also introduced a set of actions capable of dynamically modify agents mental components, which proved to be very useful for dynamic environments. In addition, we have shown that through its lifetime an agent will be safe of inconsistency.

There exists previous works that relate cognitive agents to argumentation frameworks [10,11,1]. However, none of these approaches propose an APL, they present agent architectures. Therefore, their main focus is in the impact of argumentation into deliberation, putting little attention to the dynamics of the model. For example, these works do not specify how actions modify agent mental components, or what happens with the arguments for an achieved desire. In this work, we address these issues presenting semantics rules for actions and deactivating arguments for achieved goals.

In [9], a modification of 3APL to use Dynamic Logic Programming (DLP) was introduced. Like us, they propose to use a more powerful representation language and reasoning mechanism in APLs. However DLP and DeLP are different formalisms. DLP is based in Answer-Set programming and its aim is knowledge update. DeLP is based on defeasible argumentation and its aim is to build arguments and reason with them. In DLP the decision criterion between contradictory information is fixed (newer information is preferred). DeLP uses a modular argument comparison criterion to decide between conflicting arguments. Thus, it can be programmed to prefer arguments built using newer rules, similarly to DLP.

As future work, we plan to study different goal types and determine their impact in the dynamics of the arguments. We also want to integrate an argumentative planning system and provide tools for modeling agent reactive behaviors.

References

1. Amgoud, L., Devred, C., Lagasquie, M.: A constrained argumentation system for practical reasoning. In: AAMAS, pp. 429–436 (2008)
2. Bench-Capon, T.J.M., Dunne, P.E.: Argumentation in artificial intelligence. Artif. Intell. 171, 619–641 (2007)
3. Bordini, R.H., Wooldridge, M., Hübner, J.F.: Programming Multi-Agent Systems in AgentSpeak using Jason. John Wiley & Sons, Chichester (2007)
4. Bratman, M., Israel, D., Pollack, M.: Plans and resource-bounded practical reasoning. In: Philosophy and AI: Essays at the Interface, Cambridge, Massachusetts, pp. 1–22 (1991), http://citeseer.ist.psu.edu/bratman88plans.html
5. Dastani, M., van Riemsdijk, M.B., Meyer, J.J.C.: Programming multi-agent systems in 3apl. In: Multi-Agent Programming, pp. 39–67 (2005)
6. García, A., Rotstein, N., Tucat, M., Simari, G.: An argumentative reasoning service for deliberative agents. In: Zhang, Z., Siekmann, J.H. (eds.) KSEM 2007. LNCS (LNAI), vol. 4798, pp. 128–139. Springer, Heidelberg (2007)
7. Garcia, A., Simari, G.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming 4, 95–138 (2004)
8. Kakas, A.C., Mancarella, P., Sadri, F., Stathis, K., Toni, F.: Computational logic foundations of kgp agents. J. Artif. Intell. Res. (JAIR) 33, 285–348 (2008)
9. Nigam, V., Leite, J.: Adding knowledge updates to 3apl. In: Bordini, R.H., Dastani, M.M., Dix, J., El Fallah Seghrouchni, A. (eds.) PROMAS 2006. LNCS (LNAI), vol. 4411, pp. 165–181. Springer, Heidelberg (2006)
10. Rahwan, I., Amgoud, L.: An argumentation based approach for practical reasoning. In: AAMAS, pp. 347–354 (2006)
11. Rotstein, N.D., Garcia, A.J., Simari, G.R.: Reasoning from desires to intentions: A dialectical framework. In: AAAI Conference on Artificial Intelligence, pp. 136–141 (2007)

Studying the Emergence of Money by Means of Swarm Multi-agent Simulation

Wildener M. Rodovalho, Cássio Dener N. Vinhal, and Gelson da C. Júnior

Federal University of Goiás, Goiânia GO 71605-220, Brazil
wildener@gmail.com
{cassio,gcruz}@eee.ufg.br

Abstract. The replication of scientific experiments with humans, now as artificial agents in computer simulations, under similar conditions, has shown some intriguing results in several areas, such as agent-based computational economics (ACE). This paper describes the research involving an analysis of the Kiyotaki-Wright (KW) economic model, which was created to explain the emergence of money from an exchanging environment. It is one of the most investigated models due to its simplicity and effectiveness. Following an alternative approach, the focus of this research is the use of simple intelligent agents, which use reinforcement learning, within the Swarm framework. The results obtained are quite intriguing and closer to the KW model's mathematical equilibrium than the results obtained in previous studies in scientific literature.

Keywords: Multi-agent simulation, Swarm Simulation Platform, Kiyotaki-Wright model, Reinforcement learning.

1 Introduction

There are many economical models that explain the phenomenon of the emergence of money, emphasizing the importance of a medium of exchange in commercial transactions. Among other models, there are those of Menger [1], Alchian [2], Kiyotaki-Wright (KW) [3] and Yasutomi [4]. In the present work, the KW model is used due to its simplicity, effectiveness and dissemination among scientists.

The model consists of an environment, where three types of agents are in communication and exchange goods towards the objective of obtaining a specific desired good. Each agent produces a good that is not its own desired good. They are thus required to interact with other agents and exchange goods. From these exchanges, a good (medium of exchange) emerges as money.

Many researchers tried to verify the money emergence phenomenon, as predicted by the KW model. The first simulation experiments were performed using humans, as described by Ochs [5]. These experiments were costly, used complex control protocols and a small number of human agents which did not behave as prescribed by the KW model. Later, an important step was taken during further

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 296–305, 2010.

simulation experiments, where intelligent software agents were used as a substitute for human agents, as described by Duffy [6], pointing out that software agents behaved just like human agents.

Duffy's experimental results motivated research conducted and reported by Rouchier [7], who states that due to foundational problems, she tried unsuccessfully to replicate Duffy's experiments. Rouchier also points out that Duffy did not explain clearly the rationalization mechanisms and learning strategies used by intelligent agents. Some of the questions raised are elucidated in a recent article published by Duffy [8], where learning and rationalization methods are described, considering the evolution of multi-agent paradigms.

The present research's objective is to develop a computer simulation of the KW model, to investigate the details of the agents' movements and decisions, making them clear and consistent. The approach differs from previous research by using simple intelligent agents, employing reinforcement learning [9], within the Swarm Simulation Platform [10], which is commonly used by the agent-based computational economics (ACE) research community.

The Swarm Simulation Platform used here allows the fast implementation of simple and effective multi-agent simulators and programming code reuse. It also provides a standard environment for future research and comparison of agents learning and rationalization algorithms [11].

This introduction is followed by a short description of the KW model, as well as of previous researcher's simulation experiments. Then, the KW model simulation is discussed in detail. Finally, the results obtained are presented and compared with those of the aforementioned previous researchers.

2 The Kiyotaki-Wright (KW) Model

The KW model [3] is composed by a set of three different types of agents which are paired up to enable exchange of their goods. The good produced by each agent is not the good it consumes, thus an agent must negotiate with other agent(s) to obtain its desired consumer good. It may happen that an agent receives a good y that does not correspond to its desired consumer good. This type of exchange occurs when the good, albeit undesirable, is seen by the agent as of sufficient market value to enable a future exchange for its desired consumer good. This good y is called by Kiyotaki and Wright as a medium of exchange.

Mathematically, there are agents of type $i =$ I, II e III. Each agent produces a good j, where $j = (i+1) \bmod 3$[1] Thus, for $i =$ I, II, III, one has goods $j = 2, 3, 1$, respectively[2]. According to the model, a i agent consumes good j, if $j = i$.

Each good has a storage cost c_j. When an agent receives its consumer good in an exchange, it receives an incentive u for the successful exchange. A successful exchange entails the following event sequence: the agent i receives $u - c_j$, where

[1] Another KW model exists, where $j = (i + 2) \bmod 3$. For i= I, II, III, we have thus $j = 3, 1, 2$. This model is not used nor discussed here.

[2] In order to make it clear, the agent types are represented by Roman numerals, I, II and III, while the goods are represented by Arabic numerals, 1, 2 and 3.

j is the good previously stored by the agent[3], the agent i should consume the good i received and immediately produce a good j. The relationship between each goods' cost c_j and the incentive u is expressed as: $0 < c_1 < c_2 < c_3 < u$.

A discount factor variable β, where $\beta \in (0.1)$, is used by the model to determine the end of transactions. For example: if $\beta = 0.8$, the game has an 80% chance of proceeding.

The main idea is to have each agent maximizing its gains by executing successful exchanges. An exchange only occurs when both of the paired agents agree to do so. The most complex situation that an agent i may experience is the one when it is paired with an agent that has a good k, where k is a good that agent i does not produce and also does not desire ($k \neq j$ and $k \neq i$). Agent i's decision depends on using its experience (historical data of past exchanges) to determine good k's market value. A high market value means that, with good k, the agent will have higher chances of obtaining its desired consumer good in a future exchange. In such a case, the probability of the agent to decide for the exchange would be directly proportional to the good k's market value.

As all agents try to maximize their gains, the game's outcome is an equilibrium defined by the ratios p_{ij}, which is the ratio of type i agents that have good j, where $p_{ij} = 1 - p_{ik}$. An agent is told to use the *fundamental strategy* when the focus of its decisions is on the cost of each good, leading to it always choose the cheaper product. In contrast, an agent may employ a *speculative strategy*, where its focus is on the good's market value. This strategy allows the agent to execute an exchange for a more expensive good, which increases its chances of obtaining its desired consumer good.

Economists Duffy and Ochs [5] observed that type I agent's strategy varies with the value of the incentive u. The other agent types always use the fundamental strategy. Thus, the parameter values used by Duffy and Ochs for the K-W model [3] were: u = 20 or u = 100, $\beta = 0.9, c_1 = 1, c_2 = 4$ and $c_3 = 9$.

Kiyotaki and Wright [3] prove that type I agents will always use the fundamental strategy if the condition (1) below is satisfied.

$$(c_3 - c_2) > (p_{31} - p_{21})\beta u/3 \tag{1}$$

Duffy and Ochs [5] states: "This inequality says that the difference in storage cost between goods 3 and 2 exceeds the discounted expected utility benefit to player type I from storing good 3 rather than good 2". From the parameter values specified above, the fundamental strategy use condition becomes inequation (2).

$$(p_{31} - p_{21}) < \begin{cases} \frac{5}{6} \text{ if } u = 20 \\ \frac{1}{6} \text{ if } u = 100 \end{cases} \tag{2}$$

Therefore, type I agents should refuse type II agents production good (good 3), because this has a greater storage cost. However, type II agents accept type III agents production good (good 1), while type III agents reject type I agents production good (good 2). Consequently, in the fundamental equilibrium there

[3] Or, $u - c_k$, if the agent had good k before the exchange, where $k \neq i$ and $k \neq j$.

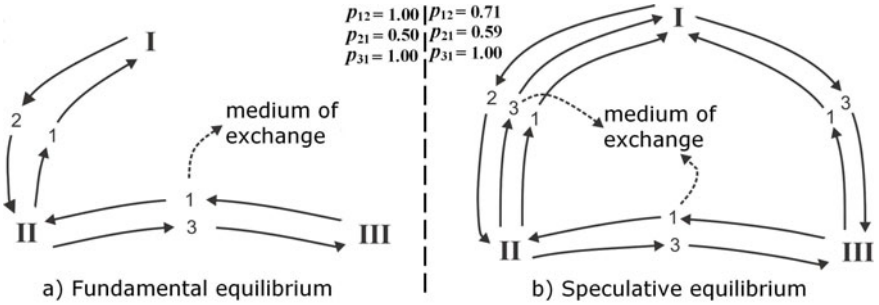


Fig. 1. Fundamental and speculative equilibria in KW model

is no interaction between agents of types I and III, as shown in Figure 1. In this equilibrium, good 1 is called a medium of exchange since it is not consumed by type II agents but is used by them to obtain good 2 from type I agents. In the fundamental equilibrium, the distribution of goods among agents converges to: $p_{12} = 1.00$, $p_{21} = 0.50$ and $p_{31} = 1.00$. This implies that $p_{31} - p_{21} = 0.50$. The only inequality in (2) that applies to this case is the one for $u = 20$. One may conclude that with these ratios, if $u = 100$, type I agents should use a speculative strategy, since the incentive is fivefold greater. In this case, the risk of storing an expensive good is compensated when the desired consumer good is obtained.

A speculative equilibrium occurs when type I agents accept good 3 produced by type II agents, despite its higher cost. With this good, they are able to make more and quick exchanges with type III agents, whence obtaining good type 1. This is explained because, for type I agents, the expectations of receiving their consumer good is higher when they are storing good type 3 than for good type 2. As type II and III agents continue acting conservatively, good 1 remains their medium of exchange. Part b of Figure 1 shows the exchanges in this equilibrium. Here the distribution of goods among agents should converge to: $p_{12} = 0.71$, $p_{21} = 0.59$ and $p_{31} = 1.00$.

3 Duffy, Ochs and Rouchier's Approaches

3.1 Duffy and Ochs' Approach

The economists Duffy and Ochs [5] undertook many experiments to investigate the validity of the KW model [3]. In their first experiment, they agents were human subjects in a computer lab, isolated from each other.

The results were satisfactory in most of the sessions. However, some players' behaviors deviated from that expected. For example, in five sessions with $u = 20$, one-third of the type I agents acted speculatively instead of using the fundamental strategy. In another five sessions, with $u = 100$, only a third of the type I agents adopted a speculative strategy and those of type III speculated even more, deviating from the model's predictions. Duffy and Ochs concluded that, due to

the information supplied to the players by their experimental system, their work induced the humans to adopt a speculative behavior.

Although he did not inform which platform he uses, Duffy's approach [6] differs from other researchers' in the manner of implementing the KW model-based agents [3]. His main objective was to determine if an agent-based computational system could be used to test the validity of the KW model. The agents' behaviors were developed based on the previous experiments with humans. Therefore, restricting the number of agents as well as the number of simulation periods.

Duffy's model [6] is entirely based on Ochs' experimental discoveries [5]. Ochs worked on simulations with a total of 18 or 24 players (evenly split among players' types). A simulation is equivalent to a session, which, in turn, is divided into games, and these into periods. One period corresponds to a simulation iteration. In each period, the agents are randomly paired for negotiations. The variable β (set at 0.90) indicates the probability of the game going on. The game ends when a certain system variable becomes greater than β.

The agents' decisions are generated with a pseudo-random number generator, with a dynamic seed. The probability that an agent adopts a strategy $s_i = 0$ is:

$$Pr[s_i^j(t) = 0] = \frac{e^{x_i^j(t)}}{1 + e^{x_i^j(t)}} \tag{3}$$

where $s_i^j(t) = 0$ means that a type I agent j does not accept an exchange of good $i + 1$ for good $i + 2$, and $x_i^j(t)$ is the difference between the benefits of storing production good $i + 1$ and that of storing good $i + 2$.

The agent's initial probability of using strategy $s_i = 0$ is 0.5, reflecting the game's initial uncertainty state. As the game progresses, the agent's successes and failures alter its strategy-use probabilities.

Reinforcement learning suggests that, if an agent is more successful when storing good $i+1$ instead of good $i+2$, then the probability of it accepting good $i + 1$ in a future exchange increases. This represents the agent's belief that its winnings can be maximized, which reflects the human player's behavior, where they change their actions based on past successes and failures. An example of this technique will be given in subsection 4.1.

Duffy noticed that his simulation results were very close to those obtained by Ochs [5]. They both used the same parameters and each agent started out with its own production good.

In the following section, Table 1 (model 1) shows the averages taken over the five sessions run by Duffy. Although bearing striking similarities to Ochs' results [5], type I and type III agents did not behave as expected.

Duffy's individual analysis indicated that some type I agents used the fundamental strategies throughout the simulation. Other agents, after they learned to speculate, speculated to the end. A significant factor is that this polarized behavior was set after the agent's first exchange experiences. Incentives received at the beginning led the agent to choose it's strategy and stick with it during the rest of the game. Duffy attributes this behavior to the agents' heterogeneity and to the incentives that the agents received (represented by u) [6].

3.2 Rouchier's Models

Ignoring implementation paradigm details, Juliette Rouchier's research [7] strove to replicate Duffy's simulation results [6]. The differing results lead Rouchier to discuss the agents' rationale and the manner in which the results are described. Rouchier [7] implements six rationale models: two duplicate Duffy's [6] and the remaining four are variations of those two. For clarity's sake, the models are numbered 1 to 6 in the following discussion.

Model 1 is an exact Duffy's duplicate. Model 2 is equivalent to one of Duffy's variations, where type II and III players are automated. Their behaviors become those described by the KW model [3], i.e., they always use the fundamental strategy. As autonomy is intrinsically an agent's characteristic, these players are not agents.

The remaining four models are Rouchier's variations of the first two models. For each of both previous models, Rouchier developed two versions. For the first version (models 3 and 5), she changed the agents' exchange process. Once an agent has exchanged good $i+1$ for good $i+2$, the reverse exchange is not allowed anymore. These agents are called "stable rational agents". In model 3, all players are also agents, i.e., they act autonomously. In model 5, type II and III players are automated, i.e., their actions follow rules predefined by the KW model [3].

For the second variant (models 4 and 6), Rouchier changes the meaning of the indicator function $I^f_{(i+1)}$ which counts the failures (f) of an agent i, having good $i+1$, in obtaining its consumer good. In Duffy's model, that variable is incremented for each period when an agent fails to obtain its consumer good, regardless of which good they are matched with. In Rouchier's version, this increment is conditional: to be qualified as a failure, the agent being negotiated with, must have the desired consumer good. Rouchier calls these agents "var-rational agents". In model 4, all the players are autonomous, while in model 6, only type I players are autonomous. Variable $I^f_{(i+2)}$ is defined in a similar way.

4 The Proposed Model and Results Thereby Obtained

This section presents the Swarm-based computational system used to implement a model similar to Duffy's [6], and employing Rouchier's [7] contributions. The results obtained are shown and commented upon.

4.1 A System Based on Duffy and Rouchier's Work

According to Rouchier [7], the model used here was divided into six types described in the previous section. The number of agents is equal to 24. However, in some simulations, this number was increased for additional experiments.

An ancillary entity called a mediator acts as an interface among agents, as well as a transaction recorder. At the beginning of each period, the mediator organizes the players in pairs randomly. Then, each agent decides if it will proceed with the exchange. For each period, the utility gain and the cost of opportunity for goods $i+1$ and $i+2$ are updated and form the agent's experience (memory).

The game's ending condition is configured at the system's highest layer. The game ends when a given variable's value exceeds β. The system is based on the Swarm Platform [10], which allowed the separation, onto different layers, of the agents involved in the exchanges from those responsible for managing the system, such as the observers which extract relevant data from the simulation. The observer agents are the first to be allocated in memory. This creates an instantiation of the moderator agent and of each agent participating in the simulation. Then, the observer agents command the moderator to start the simulation.

The moderator divides randomly the agents into pairs and sends each a message, requesting their decisions regarding the possible exchange. These agents are aware of the good offered by its pair and their own experience built upon knowledge of past exchanges. This historical knowledge has the form of the utility gain and cost of opportunity values of each good ($i+1$ and $i+2$). An agent's experience increases/decreases the probability of the agent desiring its pair's good. If both paired agents agree upon an exchange, it comes about. Otherwise, they keep their current goods and wait for the next period. The moderator is made aware of which agent pairs wish to go through with their exchange and sends them the exchange command. Then, each agent updates its utility gain and cost of opportunity values.

The observer agents record the period's data: agent pairing, pre and post-period scores, exchanging agent pairs, the distribution of goods among the agent types and the frequency at which good $i+1$ is offered in exchange for good $i+2$.

A given system variable is now compared to β. If it is greater, the simulation ends. Otherwise, a new period begins. At the end, the observers reveal a global view of the simulation, as well as details of what went on with each agent. The KW model [3] lends itself more naturally to a multi-agent representation, since each of the model's players can be described as a system agent.

The reinforcement learning technique [9] is used by the agents to refine their decisions at every period, seeking gain maximization. As described before, the initial probability ruling the choice of goods is 0.5, corresponding to complete confusion. A simple example would be the case of an agent A, in its initial state, having good $i+1$ and paired with another agent B, which has agent A's consumer good. Logically, A will suggest a trade. If B agrees, agent A will consume the received good, store a generated good $i+1$ and increase the score of that good, since it was instrumental to A obtaining its desired consumer good. This corresponds to a successful trade. On the other hand, if B does not agree with the trade, it will not happen. Then, A decreases the score related to good $i+1$, associating having that good to the unsuccessful trade.

In the first case, agent A tends to prefer good $i+1$ to good $i+2$. In the second case, the opposite occurs. According to the reinforcement learning paradigm, the initial possession of good $i+1$ is agent A's attempt to obtain its consumer good. The environment response, represented by agent B's response, indicates whether or not A was successful. Reacting to B's decision, agent A adjusts its variables in the expectation of reaching a more favorable outcome in future periods. It is noteworthy that the probability of agent B preferring good $i+1$ rather than

$i + 2$, i.e., to use the fundamental or speculative strategy, never reaches 100%, according to equation (3). This means that the surprise effect can always happen, but it is under control because that probability does not change so quickly with the time. Another fact is that, except when the agent is in face of its consumer good, its decision process is based on a random number generator function which result is compared to the probability (3).

Swarm was chosen because it is specially designed for simple models and for simple reasoning agents. It is a well-known, widespread and powerful simulation framework [12]. The model just described - a reinforcement learning and Swarm multi-agent-based KW model - generated relevant results.

4.2 Results Obtained and Comparisons with Those of Duffy and Rouchier

On average, twenty simulations were executed for each model type. Each simulation ran for an average of 100 periods and involved 24 players. For some models, additional simulations were carried out to further investigate some characteristics observed in the agents' behavior. The average of the values given by each agent to its production good $i + 1$ in exchange for good $i + 2$, is given in table 1. These values are compared to those obtained by Duffy [6] and Rouchier [7]. Considering model 1, one may observe that while agent I types were not successful due to only one third having speculated, the other agent types obtained excellent results. In contrast, in the simulations described here, and likewise in Rouchier's work [7], a larger fraction of type I agents speculated, approximately two thirds. The total amount of goods offered among players was greater, including those from type III agents, good 1, which is the cheapest good. This outcome indicates that Duffy's agents have a yet unknown characteristic causing their differing behavior. One should note that Duffy designed those agents based on experiments involving human subjects. According to Rouchier, these human traits were implemented by Duffy, but not described on his paper.

The outcome of the present research slightly differs from Rouchier's and Duffy's works, in relation to the behavior shown by type II players in models 2, 5 and 6. In those models, type II and III players are automated. Analyzing each period's simulation data, one may note that, on many occasions, type II agents were not able to offer their production good 3, in exchange for good 1. It happens due to their pairing with either an agent possessing their consumer good 2 or an agent having the same good (3). As a consequence, the simulation system associated a zero value to type 2 agents' offer. One becomes curious about how the cited authors implemented their offer value calculations. It is believed that their methods are those described by Rouchier. To evaluate that hypothesis, further simulations with more agents (30, 36, 42 and 54 agents) were made. The results obtained indicate that, as the number of agents increased, the offer values converged to 100%. That outcome is to be expected, since the greater the number of agents involved in the simulation, the greater the probability of an agent being paired with another in possession of good 1.

Table 1. Average values of speculation (%) compared

Model	Who	Type I agents		Type II agents		Type III agents	
		1^{st} half	2^{nd} half	1^{st} half	2^{nd} half	1^{st} half	2^{nd} half
1	Duffy	19.00	32.00	77.00	99.00	22.00	4.00
	Rouchier	74.00	68.00	80.00	93.00	73.00	81.00
	This work	61.74	67.28	61.62	67.28	66.97	65.03
2	Duffy	62.00	73.00	100.00	100.00	0.00	0.00
	Rouchier	91.00	100.00	100.00	100.00	0.00	0.00
	This work	78.60	88.22	96.80	95.81	0.00	0.00
3	Rouchier	68.00	77.00	76.00	79.00	66.00	66.00
	This work	49.40	55.90	55.15	53.38	57.73	54.15
4	Rouchier	45.00	42.00	53.00	47.00	42.00	52.00
	This work	44.45	63.60	61.82	55.28	25.58	11.18
5	Rouchier	80.00	88.00	100.00	100.00	0.00	0.00
	This work	75.18	89.07	95.27	96.76	0.00	0.00
6	Rouchier	80.00	100.00	100.00	100.00	0.00	0.00
	This work	70.62	84.01	97.02	96.89	0.00	0.00

Given the three models using wholly automated players, the best results for type III players were obtained for model 4. These used the fundamental strategy and, in some cases, more speculation was observed for type I agents: up to 80%.

The best results achieved are those of the models 2, 4 and 5, because they converge to the equilibrium described by the KW model [3]. For models 1 and 3, the agents were examined individually to identify why they did not behave as expected, according to the KW model. With reinforcement learning [9], each agent increases its experience at each period. At start, an agent A depends entirely on the random number generator, which determines which decision the agent must take, when confronted with another agent B that has neither its consumer good nor its produced good. The probability of an agent behaving as expected at start is 0.5. It may happen that the agent decides to make an exchange different from that expected and, in the next period, obtains a successful trade with the stored good. This action will increase, in the remaining periods, the probability that agent A make the same kind of exchange.

Rouchier [7] takes into account the amount of agents and the time they take to assimilate the best decision. For model 1, over seven simulations with 18 agents of each type and an average of 240 periods, 71.4% of type I agents speculated.Therefore, good results were obtained for some models while others require further refining. It would have been useful to have Duffy's notes on how his agents reason, what would have allowed a closer replication of his simulations.

5 Concluding Remarks

Kiyotaki-Wright's model [3] was discussed as an abstraction of an exchange market where a good emerges as a medium of exchange. Duffy and Ochs [5] human subjects trials and Duffy's artificial agent simulations [6] confirmed the

effectiveness of the KW model. A description of these simulations became the foundation for replicating Duffy's work and for comparing with Rouchier's alternative approaches [7].

Of the six models described here, three (2, 4 and 5) obtained good results, an average of 24.5% better than Rouchier and Duffy for type I agents regarding the second half of the simulations. In the remaining three models, attempts to reach results closer to those expected were made by increasing the number of agents and the number of simulation periods. Another decisive factor was that individual scrutiny revealed that some type I agents were successful when using the fundamental strategy instead of the expected, speculative strategy.

The Swarm Simulation Platform [10] facilitated code structuring and decreased development time. While its use did not lead to better results, that was not its purpose. A key result was that human-based experiments could be reproduced with the help of a simulation toolkit.

Use of the Swarm platform will make easier to compare alternative models created to further investigate agent's behavior. The agents' reasoning process may be enhanced by supplying them with a more complete view of the market situation which will supply more favorable decision-making conditions. The interaction between the parameter values and the agents' scores for each model should be considered by future research. Finally, new models could be implemented as part of the decision process algorithm used by the agents, e.g. case-based reasoning, negotiation algorithms, etc.

References

1. Menger, C.: Principles of Economics. Ludwig von Mises Institute, Auburn (1871)
2. Alchian, A.: Uncertainty, Evolution and Economic Theory. The Journal of Political Economy 58, 211–221 (1950)
3. Kiyotaki, N., Wright, R.: On Money as a Medium of Exchange. The Journal of Political Economy 97, 927–954 (1989)
4. Yasutomi, A.: The Emergence and Collapse of Money. Physica D 1–2, 180–194 (1995)
5. Duffy, J., Ochs, J.: Emergence of Money as a Medium of Exchange: An Experimental Study. American Economic Review 89, 847–877 (1999)
6. Duffy, J.: Learning to Speculate: Experiments with Artificial and Real Agents. Journal of Economic Dynamics and Control 25, 295–319 (2001)
7. Rouchier, J.: Re-implementation of a multi-agent model aimed at sustaining experimental economic research: The case of simulations with emerging speculation. Journal of Artificial Societies and Social Simulation 6 (2003)
8. Duffy, J.: Agent-Based Models and Human Subject Experiments. In: Tesfatsion, L., Judd, K.L. (eds.) Handbook of Computational Economics, vol. 2, pp. 949–1011. Elsevier, North-Holland (2006)
9. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
10. Minar, N., et al.: The Swarm simulation system: a toolkit for building multi-agent simulations. Santa Fe Institute, Santa Fe (1996)
11. Railsback, F.S., Lytinen, L.S.: Agent-based Simulation Platforms: Review and Development Recommendations. Simulation 82, 609–623 (2006)
12. Tobias, R., Hofmann, C.: Evaluation of Free Java-libraries for Social-scientific Agent Based Simulation. Journal of Artificial Societies and Social Simulation 7 (2004)

Revisiting the Readability Assessment of Texts in Portuguese

Carolina Scarton, Caroline Gasperin, and Sandra Aluisio

Center of Computational Linguistics (NILC), University of São Paulo
Av. Trabalhador São-Carlense, 400. 13560-970 – São Carlos/SP, Brazil
{carol.scarton,cgasperin}@gmail.com,
sandra@icmc.usp.br

Abstract. The Web content accessibility guidelines (WCAG) 2.0 include in its principle of comprehensibility an accessibility requirement related to the level of writing. This requirement states that websites with texts demanding higher reading skills than individuals with lower secondary education possess (fifth to ninth grades in Brazil) should offer them an alternative version of the same content. Natural Language Processing technology and research in Psycholinguistics can help automate the task of classifying a text according to its reading difficulty. In this paper, we present experiments to build a readability checker to classify texts in Portuguese, considering different text genres, domains and reader ages, using naturally occurring texts. More precisely, we classify texts in simple (for 7 to 14-year-olds) and complex (for adults), and address three key research questions: (1) Which machine-learning algorithm produces the best results? (2) Which features are relevant? (3) Do different text genres have an impact on readability assessment?

1 Introduction

WCAG 2.0 include in its principle of comprehensibility an accessibility requirement related to the level of writing in Web applications (success criterion 3.1.5). This requirement states that websites with texts demanding higher reading skills than individuals with lower secondary education possess should offer them an alternative version of the same content or supplementary content intended for them [1]. For instance: a text summary with simplified content; illustrations, pictures, and symbols to facilitate understanding; audio versions of texts, among others.

Currently, the task of selecting automatically appropriate reading material for a particular grade level or target audience is receiving a great deal of attention, especially now that the Web is a huge source of material for teachers and students. Web search engines are very successful in selecting a topic of interest, and readability checkers should be as good in selecting texts according to their readability level. Moreover low literacy readers will bring their special needs to a scenario of digital inclusion and accessibility.

Traditional readability metrics, such as the Flesch index [2] (based on the average number of syllables per word and the average number of words per sentence in a passage of text), are used today in many settings, since they are very easy to compute.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 306–315, 2010.

The results they produce, however, do not always represent the complexity of a text. NLP technology and research in Psycholinguistics can provide a more sophisticated way to check text readability and consequently to automate the task of classifying a text according to grade levels or specific audiences.

The majority of the studies deal with the English language. English has a long history in readability metrics – 200 formulas have been reported from 1920 to 1980 [3] – and recent work in Natural Language Processing (NLP) has shown the benefit of using statistical language processing techniques for reading level assessment [4-15]. As far as we are aware, there are only two readability tools available for Portuguese: an adaptation of the Flesch Reading Ease index for Portuguese [16], and more recently a readability checker based on modern approaches to readability and text simplification for readers with low levels of literacy [17].

The work of Martins et al. [16] marked the beginning of readability assessment studies for Portuguese, mainly because of their effort to build a corpus of texts of different readability levels: texts appropriate for primary education students, high school and college students, and also graduate students. Their system was integrated into Microsoft Word for Portuguese. Aluisio et al. [17] investigate machine-learning models to identify the readability levels of original and manually simplified texts.

We aim to build on the steps of the work by Martins et al. and the recent findings of Aluisio et al. and propose a modern approach for readability assessment of naturally occurring texts in Portuguese. Our long term goal is to build a readability checker to classify Web texts in Portuguese, considering different text genres, domains and reader ages, using only naturally occurring texts and a scale similar to that by Martins et al..

As the initial step towards our goal, in this paper we present an evaluation of machine-learning algorithms for the reading level assessment of texts in Portuguese; more precisely, we classify texts in simple (for 7 to 14-year-olds) and complex (for adults). We address the following research questions: (1) Which machine-learning algorithm produces the best results for this task? (2) Which features are relevant? (3) Do different text genres have an impact on assessing readability for children and adults?

On the one hand, our work is similar to that by Martins et al. [16], who used corpora of authentic texts, although not Web texts and its new genres as our long term goal, and is also close to that by Aluisio et al. [17], who also used a comprehensive set of features and a machine-learning approach (instead of authentic texts, Aluisio et al.'s sample of simple texts is composed of manually simplified content). On the other hand, we model our problem with different classes and use naturally occurring texts, diverging from those studies in fundamental points.

In the remainder of this paper, we discuss prior work on readability assessment (Section 2), and then present our features, corpora and modeling techniques (Section 3), and the experiments performed to answer our research questions (Section 4). Section 5 summarizes our first contributions and indicates future work.

2 A Summary of Readability Assessment Studies

Lately, research on the assessment of text readability for the English language has concentrated on five topics: (i) the set of features used to capture the various aspects

of readability, so as to evaluate the contribution of lexical, syntactic, semantic, and discursive features; (ii) the audiences to which the text readability measurement is intended; (iii) the effects of text genres on the calculation of text difficulty; (iv) the most appropriate type of learning technique, i.e., nominal, ordinal or interval scales of measurement; and (v) the development of applications to assess reading difficulty automatically.

With regard to applications – the last topic – Miltsakaki and Troutt [5, 6] proposed an automatic tool to evaluate the reading difficulty of Web texts in real time, designed for teenagers and adults with low literacy skills. On the other hand, Newbold and Gillam [8] focus on a desktop application, implementing an extension to the Open Office text processor that considers text cohesion, propositional density, and word familiarity. The fourth topic under discussion was investigated by Heilman et al. [13]; they compared the effectiveness of machine-learning models to develop an appropriate measurement scale for reading difficulty. The key point was to compare models according to the way they view the readability scale: as a nominal, ordinal or interval set of levels. They concluded that an ordinal scale is the best way to model readability levels. Aluisio et al. [17] present a similar experiment considering original and manually simplified texts, with similar results. Another study focusing on the same topic is that by Petersen and Ostendorf [14], in which classification and regression techniques were employed to predict readability scores.

As for topic (i), a unified framework was proposed to measure text quality [7], resembling the composition of rubrics used for essay scoring [9]. This single framework included the following features: vocabulary, syntax, elements of lexical cohesion, entity coherence, and discourse relations. Concerning specific target audiences, Heilman et al. [12] and Schwarm and Ostendorf [15] address the readability assessment of texts for learners of English as a second language. As regards the intellectually disabled and the cognitively impaired by Alzheimer's disease, Feng et al. [11] and Roark et al. [4], respectively, stand out. Text genres, topic (iii), were studied by Sheehan et al. [10], who focused on models for literary and expository texts, since the traditional metrics tend to overpredict the difficulty of literary texts and underpredict the difficulty of expository ones.

Our research seeks to contribute to the field in the following: (i) a set of features with cognitively motivated metrics is employed to explain better the difficulty of a text; (ii) a different audience is intended (children of different ages and adults); (iii) the readability of multiple text genres together can be assessed and the Portuguese language is our aim.

3 Corpora and Features for Readability Assessment

Table 1 shows some statistics about the four main corpora considered in this paper, derived from the following sources: newspaper *Zero Hora* (ZH),[1] section "Para seu filho ler" (PSFL),[2] magazine *Ciência Hoje* (CH),[3] and magazine *Ciência Hoje das*

[1] http://zerohora.clicrbs.com.br/
[2] http://wp.clicrbs.com.br/editor/2010/04/15/noticias-feitas-para-seu-filho-ou-aluno-ler/
[3] http://cienciahoje.uol.com.br/revista-ch

Crianças (CHC).[4] Two other smaller corpora were also used to diversify the sources of the two genres: 50 texts from "Caderno Ciência" (CC), a section on science popularization of the newspaper *Folha de S. Paulo*,[5] published from 1994 to 2005; and 50 news texts from the newspaper *Diário Gaúcho* (DG),[6] dedicated to social classes C and D, published in 2008.

The corpus ZH contains 166 news texts written in 2006 and 2007. In this work, we consider the corpus ZH as "complex", for its texts were written for adults. "Para seu filho ler" (PSFL) is a section of the newspaper *Zero Hora*, which features some news on the same topics as in *Zero Hora*, but in a form appropriate for children aged 8 to 11 years. In this study, the corpus PSFL is considered "simple". The corpus CH is made up of 130 popular science texts from issues of the magazine *Ciência Hoje* published in 2006, 2007, and 2008. This corpus is considered "complex" as well. The corpus CHC is composed of 127 popular science texts from issues of the magazine *Ciência Hoje das Crianças* published in 2006, 2007, and 2008, whose targeted readers are children between 7 and 14 years old. This corpus is considered "simple".

Table 1. Statistics of the main corpora considered in this paper

Corpus	Number of texts	Number of words	Average number of words per text	Age group
ZH	166	63,996	385.518	Adults
CH	130	81,139	624.146	Adults
PSFL	166	19,257	116.006	8-11-year-olds
CHC	127	56,096	441.701	7-14-year-olds
CC	50	16,530	330.600	Adults
DG	50	4,482	89.64	Adults

As features to assess text readability, we have used the Coh-Metrix-Port tool [18], composed of 48 psycholinguistic metrics, based on the tool Coh-Metrix (for the English language) [19-21]. Coh-Metrix-Port metrics are presented below:

- Basic count: number of words, number of sentences, number of paragraphs, sentences per paragraph, words per sentence, syllables per word, incidence of verbs, incidence of nouns, incidence of adverbs, incidence of adjectives, incidence of pronouns, incidence of content words (nouns, adjectives, adverbs, and verbs) and incidence of function words (articles, prepositions, pronouns, conjunctions, and interjections).
- The Flesch index.
- Constituents: incidence of noun phrases, modifiers per noun phrase, and words before main verbs.
- Frequencies: frequency of content words and minimum frequencies of content words.

[4] http://www.chc.org.br/
[5] http://www.folha.uol.com.br/
[6] http://www.clicrbs.com.br/especial/rs/diario-gaucho/

- Connectives: incidence of all connectives, incidence of positive additive connectives, incidence of positive temporal connectives, incidence of positive causal connectives, incidence of positive logical connectives, incidence of negative additive connectives, incidence of negative causal connectives, incidence of negative temporal connectives, and incidence of negative logical connectives.
- Logical operators: incidence of logical operators, number of *e* [and], number of *ou* [or], number of *se* [if], and number of negations.
- Pronouns, types and tokens: incidence of personal pronouns, pronouns per phrase, and type/token relation.
- Hypernyms: verb hypernyms.
- Ambiguities: verb, noun, adjective, and adverb ambiguity.
- Coreferences: argument overlap, argument overlap in adjacent sentences, word root overlap, word root overlap in adjacent sentences, content word overlap in adjacent sentences.
- Anaphoras: anaphoric reference and adjacent anaphoric reference.

4 Experiments

Here, we address the following research question: (1) Which machine-learning algorithm produces the best results for our task? Section 4.2 shows our experiments with nominal classifiers IBk (K = 7), J48, JRip, MultilayerPerceptron, SMO, and NaiveBayes. In order to answer our second research question, (2) Which features are relevant?, we experiment with feature selection algorithms; this is described in Section 4.1. The worst-case analysis is discussed in Section 4.3. Finally, Section 4.4 presents our experiments to answer our third question, (3) Do different text genres have an impact on readability assessment?

4.1 Feature Selection

For this paper, we conducted experiments in three scenarios: (i) with all features; (ii) eliminating redundant features manually and then using the Information Gain feature selection algorithm (InfoGainAttributeEval implementation from the Weka package [22]); and (iii) eliminating redundant features manually and then using the SVM (Support Vector Machine) feature selection algorithm (SVMAttributeEval implementation from the Weka package). The redundant features that were eliminated manually are: number of words, number of sentences, syllables per word, and words per sentence, which are included in the Flesch index feature; and the number of paragraphs, which is implicit in the metric sentences per paragraph. The Information Gain algorithm was chosen to be tested because of its popularity, and the SVM algorithm was chosen in order to allow a clearer comparison between the performance of the classification models other than SVM and the SVM classifier performance (Section 4.2).

The InfoGainAttributeEval eliminated seven features: incidence of negative temporal connectives, ambiguity of adverbs, ambiguity of nouns, modifiers per noun phrase, incidence of negations and incidence of e. On the other hand, the SVMAttributeEval eliminated 10 features (the last features of the ranking): adjacent anaphoric reference, anaphoric reference, ambiguity of verbs, incidence of e, incidence of logical operators,

incidence of negative logical connectives, incidence of noun phrases, incidence of content words, incidence of negative additive connectives and ambiguity of adverbs.

4.2 Classification

As we are considering only two classes, we see our task as a classification problem (we do not consider the ordinal classification, pointed out as the most appropriate paradigm to deal with text complexity [12], since, with only two classes, both models produce the same results). Given the scenarios introduced in Section 4.1, we conducted our experiments using eight machine-learning algorithms: K-NN (with k = 7, using Euclidean distance)[7], NaiveBayes, J48, JRip, MultilayerPerceptron (MP), and SMO (with PolyKernel with p = 1.0). All algorithms are implementations from the Weka package [22]. For all algorithms, we used the 10-fold cross-validation procedure. Table 2 shows the values of F-measure and Minimum Absolute Error (MAE) for all algorithms in our three scenarios.

As we can see in Table 2, the best result in scenario (i), both considering F-Measure and MAE, was that of the SMO, as expected, since this algorithm is widely cited in the literature about text intelligibility as the one that performs better. The second best result was that of MultilayerPerceptron. The worst results were those of J48.

Table 2. Results of algorithms in scenarios (i), (ii) and (iii)

	Scenario (i)		Scenario (ii)		Scenario (iii)	
	F-Measure	MAE	F-Measure	MAE	F-Measure	MAE
7-NN	0.915	0.124	0.877	0.157*	0.88*	0.156*
J48	0.886*	0.117*	0.891	0.114	0.891	0.116
SMO	**0.944**	**0.056**	**0.937**	**0.063**	**0.941**	**0.059**
MP	0.935	0.066	0.924	0.08	0.924	0.078
JRIP	0.915	0.108	0.92	0.108	0.905	0.122
NaiveBayes	0.925	0.08	0.875*	0.085	0.929	0.075

In scenario (ii), we can see that the SMO algorithm was also the one with the best F-measure and MAE. In this case, the worst result was the F-measure in NaiveBayes and the MAE in 7-NN with Euclidean distance.

In spite of believing that feature selection improves the performance of classification algorithms, we see that the results for F-measure and MAE in scenario (i) (with all features) are better than those in scenario (ii) (with feature selection – InfoGainAttributeEval). We suppose these results can be justified by a possible overfitting when we use all features.

Lastly, the third column of Table 2 shows the results for scenario (iii). In this scenario, the SMO was again the best algorithm in terms of F-measure and MAE. The worst results are those of 7-NN. This scenario has not outdone the performance of scenario (i) either.

[7] We tested k values from 1 to 10 and k=7 achieved the best results.

It is known that the SVM algorithm (SMO) already has the capacity for selecting the most relevant features, and that this factor possibly contributes to its better performance. Moreover, the fact that scenarios (ii) and (iii) have not outdone the performance achieved by the several algorithms in scenario (i) can be a result of insufficient data to reach high precision in feature selection, since these processes require a large amount of data for analysis.

4.3 The Worst-Case Analysis

As we intend to distinguish texts for adults (complex) from texts for children (simple), we should pay attention to the characteristics of this task. It is necessary to define the worst type of errors in the classification and assess the performance of the algorithms from this point of view as well.

In this work, the worst case is to have "complex" texts classified as "simple". The problem is that if a child tries to read a text intended for adults, that child will probably not understand it, and the classifier user might be deceived. With this problem in mind, we analyzed the classification algorithms according to the rate of false positives for the "simple" class. Table 3 shows the rate of false positives for the "simple" class (FP-simple: number of false positives in relation to all instances classified as simple), for all algorithms in the three scenarios. The algorithm that shows the best result in all scenarios is 7-NN (0.03 for scenario (i), 0.041 for scenario (ii) and 0.03 for scenario (iii)). It means that, considering FP-simples rate, K-NN with K=7 performs better than SMO. We can conclude that the worst-case analysis help us to define the best algorithm for classifying texts for children.

Table 3. Results of false positives in the "simple" class

	FP –simple		
	(i)	(ii)	(iii)
7-NN	**0.03**	**0.041**	**0.03**
J48	0.098	0.088	0.095
SMO	0.051	0.061	0.061
MP	0.061	0.074	0.068
JRIP	0.111	0.088	0.068
NaiveBayes	0.057	0.078	0.152

4.4 Genre and Domain versus Readability Classification Performance

To test whether different genres and domains have an impact on classification, we created two independent classifiers – one trained only with news texts (ZH + DG + PSFL) and the other trained only with popular science texts (CH + CC + CHC) – and assessed how well each one performed to separate texts for adults from texts for children. We also analyzed the worst possible case to find out whether there were many errors in the classification of complex texts as simple ones. Both classifiers were developed using the SMO algorithm, with the same parameters described in Section 4.2. We chose the SMO because it produced the best results for F-measure and MAE in the experiments reported in Section 4.2.

Table 4 shows the five main features in each classifier (according to the InfoGain-AttributeEval algorithm). The five metrics that influenced the most the classification are similar in both genres (except for "Incidence of pronouns", which appears only in ZH + DG + PSFL, and "All connectives", which appears only in CH + CC + CHC). What has changed was their relevance. After building these models, we tested the classifiers on texts of different genres: the ZH+DG+PSFL classifier was tested on the CH + CC + CHC data, and vice versa.

Table 4. Main features of classifiers per genre

ZH + DG + PSFL	CH + CC + CHC
Pronouns per phrase	Flesch
Incidence of nouns	Incidence of verbs
Flesch	All connectives
Incidence of verbs	Incidence of nouns
Incidence of pronouns	Pronouns per phrase

Table 5 shows these results. As we can see in Table 5, the classifier trained with texts from the corpora ZH, DG and PSFL, when assessed against texts of a different genre, produced the lowest rate of false positives for the "simple" class.

Table 5. Results of the experiment to classify texts of a different genre

	F-Measure	MAE	FP - simple	FP – complex
ZH + DG + PSFL classifier tested on CH + CC + CHC data	0.831	0.163	**0.033**	0.346
CH + CC + CHC classifier tested on ZH + DG + PSFL data	0.854	0.147	0.204	0.072

Even if this classifier's F-measure and MAE scores are lower than for the CH+CC+CHC classifier, the values are similar and satisfactory. In this experiment, we observed that a classifier built with news texts only, besides showing satisfactory values of F-measure and MAE, also shows relatively few errors when we use it to classify texts of a different genre (popular science). In addition, the value of the worst case in the classification of popular science texts is also considerably low. One reason for that is the proximity between genres, whose main purpose is to inform, although popular science texts have also an educational function. The difference between the two corpora are the domains they deal with: whereas popular science texts have a set domain – science –, newspaper texts deal, in general, with a variety of themes, such as: daily life and community issues, environment, politics, sports, culture, among others.

5 Conclusions and Future Work

In this paper, we have shown our initial exploring experiments towards creating a readability assessment system for checking the readability level of texts in Portuguese. We have limited the case to a two-class classification problem, which consists of

indicating whether texts are complex (appropriate for adults) or simple (appropriate for children). We have tested several machine-learning algorithms on this task and also experimented with feature selection algorithms, in order to select the most relevant features from a set of 48 cognitively motivated ones extracted with Coh-Metrix-Port. Our experiments have shown that the SVM algorithm, with no feature selection preprocessing, achieved the best performance on our task.

Our training and testing corpora are composed of texts at both readability levels, complex and simple, in two different (although correlated) genres and domains: three of our corpora are news texts and the other three are popular science texts. We experimented training a classifier with corpora of one genre and testing it on the corpora of the other genre. Our results have shown that the classifier trained on newspaper texts was able to generalize and classify considerably well the popular science texts. As future work, we aim to experiment with a more fine-grained set of classes, including texts addressed to high school, college and also graduate students, using naturally occurring texts from the Web, which has become a mature source of reading material for all these classes. Moreover, novel texts genres have emerged with the Web, while others have been adapted to fit in this new environment and need to be assessed with regard to reading difficulty. A first challenge in this search will be the compilation of a diverse corpus classified by genres of interest, since this is a difficult research question in itself [23]. However, as we have seen in the experiments of Section 4.4, the proximity between genres can allow the use of a classifier trained with a specific genre on texts of another genre, making our task less painful.

Acknowledgments

We thank the Brazilian Science Foundation FAPESP for financial support.

References

1. W3C. Web content accessibility guidelines (WCAG) 2.0. W3C Recommendation (December 2008), http://www.w3.org/TR/WCAG20/
2. Flesch, R.: A new readability yardstick. Journal of Applied Psychology 32, 221–233 (1948)
3. DuBay, W.H.: The principles of readability. Impact Information, Costa Mesa (2004), http://www.impact-information.com/impactinfo/readability02.pdf
4. Roark, B., Mitchell, M., Hollingshead, K.: Syntactic complexity measures for detect-ing mild cognitive impairment. In: The Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, Prague, Czech Republic (2007)
5. Miltsakaki, E., Troutt, A.: Read-X: Automatic Evaluation of Reading Difficulty of Web Text. In: The Proceedings of E-Learn 2007, by the Association for the Advancement of Computing in Education (AACE), Quebec, Canada (2007)
6. Miltsakaki, E., Troutt, A.: Real Time Web Text Classification and Analysis of Reading Difficulty. In: The Proceedings of the 3rd Workshop on Innovative Use of NLP for Building Educational Applications, at the 46th Meeting of the Association for Computational Linguistics and Human Language Technologies, Columbus, OH (2008)
7. Pitler, E., Nenkova, A.: Revisiting readability: A unified framework for predicting text quality. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), Wikiki, Honolulu, Hawaii (2008)

8. Newbold, N., Gillam, L.: The Linguistics of Readability: The Next Step for Word Processing. In: Proceedings of the NAACL HLT 2010, Workshop on Computational Linguistics and Writing, Los Angeles, California, pp. 65–72 (2010)
9. Burstein, J., Chodorow, M., Leacock, C.: CriterionSM Online Essay Evaluation: An Application for Automated Evaluation of Student Essays. In: The Proceedings of the Fifteenth Annual Conference on Innovative Applications of Artificial Intelligence, Acapulco, Mexico (2003)
10. Sheehan, K.M., Kostin, I., Futagi, Y.: Reading Level Assessment for Literary and Expository Texts. In: McNamara, D.S., Trafton, J.G. (eds.) Proceedings of the 29th Annual Cognitive Science Society, pp. 18–53. Cognitive Science Society, Austin (2007)
11. Feng, L., Elhadad, N., Huenerfauth, M.: Cognitively Motivated Features for Readability Assessment. In: The Proceedings of Conference of the European Chapter of the Association for Computational Linguistics (EACL), Athens, Greece, pp. 229–237 (2009)
12. Heilman, M., Collins-Thompson, K., Callan, J., Eskenazi, M.: Combining lexical and grammatical features to improve readability measures for first and second language texts. In: The Proceedings of NAACL HLT, Rochester, NY, pp. 460–467 (2007)
13. Heilman, M., Collins-Thompson, K., Eskenazi, M.: An Analysis of Statistical Models and Features for Reading Difficulty Prediction. In: Proceedings of the 3rd Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, Rochester, NY, pp. 71–79 (2008)
14. Petersen, S.E., Ostendorf, M.: A machine learning approach to reading level assessment. Computer Speech and Language 23, 89–106 (2009)
15. Schwarm, S.E., Ostendorf, M.: Reading Level Assessment Using Support Vector Machines and Statistical Language Models. In: The Proceedings of the 43rd Annual Meeting of the ACL, pp. 523–530. University of Michigan, Ann Arbor (2005)
16. Martins, T.B.F., Ghiraldelo, C.M., Nunes, M.G.V., Oliveira Junior, O.N.: Readability formulas applied to textbooks in Brazilian Portuguese. ICMC Technical Report, N. 28, p. 11 (1996)
17. Aluisio, S.M., Specia, L., Gasperin, C., Scarton, C.: Readability Assessment for Text Simplification. In: Proceedings of the NAACL-HLT 2010, Workshop on Innovative Use of NLP for Building Educational Application, Los Angeles, USA, pp. 1–9 (2010)
18. Scarton, C.E., Aluísio, S.M.: Análise da Inteligibilidade de textos via ferramentas de Processamento de Língua Natural: adaptando as métricas do Coh-Metrix para o Português. Revista Linguamática (Revista para o Processamento Automático das Línguas Ibéricas) 2(1), 45–61 (2010)
19. Graesser, A.C., McNamara, D.S., Louwerse, M.M., Cai, Z.: Coh-Metrix: Analysis of text on cohesion and language. Behavioral Research Methods, Instruments, and Computers 36, 193–202 (2004)
20. McNamara, D.S., Louwerse, M.M., Graesser, A.C.: Coh-Metrix: Automated cohe-sion and coherence scores to predict text readability and facilitate comprehension. Grant proposal (2002), http://cohmetrix.memphis.edu/cohmetrixpr/publications.html
21. Crossley, S.A., Louwerse, M.M., McCarthy, P.M., McNamara, D.S.: A linguistic analysis of simplified and authentic texts. Modern Language Journal 91, 15–30 (2007)
22. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
23. Santini, M.: Characterizing Genres of Web Pages: Genre Hybridism and Individualization. In: Hawaii International Conference on System Sciences, 40th Annual Hawaii International Conference on System Sciences (HICSS 2007), p. 71 (2007)

Improved Text Generation Using N-gram Statistics

Eder Miranda de Novais, Thiago Dias Tadeu, and Ivandré Paraboni

School of Arts, Sciences and Humanities, University of São Paulo (USP / EACH)
Av. Arlindo Bettio, 1000 - São Paulo, Brazil
{eder.novais,thiagoo,ivandre}@usp.br

Abstract. In Natural Language Generation (NLG) systems, a general-purpose surface realisation module will usually require the underlying application to provide highly detailed input knowledge about the target sentence. As an attempt to reduce some of this complexity, in this paper we follow a traditional approach to NLG and present a number of experiments involving the use of n-gram language models as an aid to an otherwise rule-based text generation approach. By freeing the application from the burden of providing a linguistically- rich input specification, and also by taking some of the generation decisions away from the surface realisation module, we expect to make NLG techniques accessible to a wider range of potential applications.

Keywords: Text Generation, Surface Realisation, Language Modelling.

1 Introduction

Natural Language Generation (NLG) systems enable non-linguistic data to be visualised as text reports. Systems of this kind often follow a pipelined architecture as proposed in [1], in which content messages are extracted from the domain data and are subsequently enriched with linguistic information, up to the point in which a full (and usually language-independent) abstract sentence specification is obtained. In the final stages of the NLG pipeline, sentence specifications are assigned content words and then linearised, in the task known as surface realisation [2,3].

Surface realisation takes as an input a semantic representation of *what* to say, and determines *how* to say it, producing output surface strings in a given target language. A general-purpose surface realisation module will usually require a highly detailed, linguistically-motivated input specification to be provided by the underlying application. For instance, in order to generate a sentence as 'The foreign students will review this paper', a surface realisation system as [4] will normally expect as an input a feature structure as the following simplified example[1]:

[1] Some of the specified features in this example (e.g., action person) may be required in highly-inflected languages such as Portuguese, but this may not necessarily be the case in other languages, e.g., English.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 316–325, 2010.

```
sentence(
           voice=active,
           agent(   head=student, det=definite,
                    gender=masc, number=plural, premodifier=foreign),
           action(  head=review, tense=future,
                    mode=indicative, person=3p),
           patient( head=paper, det=demonstrative,
                    dettype=this, gender=masc, number=singular)
       )
```

In a wide-coverage surface realisation system, such rich representation is compulsory if we are to have full control over the output text. However, unless the underlying application is linguistically-oriented by design, it may be difficult to provide information in this level of detail for a number of reasons. First, this representation assumes that all content words are known in advance, that is, lexical choice has already been performed and the surface realisation module is set to use, e.g., the word 'student' (and not 'pupil' or 'learner' etc.) to represent the agent head concept.

Second, this representation also assumes that the agent property 'foreign' is to be realised in a pre-modifier position, which is not straightforward even when a single modifier is involved (compare to, e.g., 'the student (that was) mentioned', in which the participle modifier appears after the concept head.)

Finally, morphologically-rich languages require nouns, adjectives, participles and others to be gender- and number-inflected. Thus, unless the application is assumed to provide this kind of information as well, the surface realisation module must provide a full grammar for each target language under consideration.

These examples highlight the fact that either the application or the surface realisation module will have to implement - at a considerable cost - a number of language-dependent generation decisions. As an attempt to overcome some of these difficulties, in this paper we follow [5] and present a series of experiments involving the use of n-gram language models as an aid to an otherwise rule-based generation approach for the Brazilian Portuguese language. In doing so, we would like to free the application from the burden of providing a full input specification (hence making it easier to adapt), and also to take some of the generation decisions away from the surface realisation module (making it easier to deploy to a new target language.)

The reminder of this paper is organised as follows. Section 2 describes the surface realisation issues that are the main focus of our experiments and related work in the field. Section 3 presents each of the experiment settings and their results. Section 4 discusses our findings, and Section 5 presents our conclusions and future work.

2 Experiments Overview

In what follows we use the surface realisation system described in [4] to investigate the role of n-gram language models as a means to simplify the system input

specification, and also as a possible substitute for some of the required grammar rules. To this end, three realisation-related problems will be taken as our working examples: lexical choice[2], ordering of nominal modifiers, and verb-complement agreement. The sentences to be generated are similar to those found in a small corpus of emails in an undergraduate project domain, conveying both questions made by students and replies sent by their project tutor.

In all experiments, our general strategy consists of leaving some of the input knowledge or rules unspecified, and using a trainable n-gram language model of Portuguese to guide the decision-making as proposed in [5]. The language models under consideration are standard 2-gram and 3-gram models[3] with back-off, built from a 40-million words corpus of Brazilian Portuguese newspapers articles. The NLG system itself, a standard template-based approach that takes feature structures as an input to produce word strings, will not be presently discussed - see [4] for details.

Our first research question concerns lexical choice, that is, the task of finding suitable words to represent each input concept given a potentially wide choice of synonyms [6]. The issue becomes particularly complex if, as pointed out in [7], we consider that sense-tagged corpora are rarely obtainable, and that the surface realisation of a given concept may have to be disambiguated not only among a potentially large number of synonyms, but among a large number of synsets in the first place.

In our experiments, two instances of lexical choice decision will be considered: those involving the realisation of the head constituents of noun phrases (NPs) and the head constituents of verb phrases (VPs.) Our goal in both cases is to investigate whether these choices can be left unspecified in the input data, leaving the decisions to be made with the aid of a n-gram language model. NP and VP head choices will be investigated separately in Experiments 1 and 2 in the next section, and also as a combined Experiment 3 which evaluates the limits of the n-gram approach to lexical choice in more complex decision-making.

Our second research question concerns the ordering of nominal modifiers, that is, the task of determining the linear order of realisation of adjectives, determiners, participles etc. attached to a noun head. The task is known to be considerably complex even for languages such as English, in which modifiers are mainly concentrated in the pre-nominal position (e.g., 'the small red book', but not '*the small book red'), and the correct orderings are not easily captured by grammar rules. Existing approaches tend to focus on the ordering of English pre-modifiers, which has been addressed as a classification problem [8,9]. We notice however that modifiers in Romance languages may often appear in post-position as well, and that they may be easily combined with pre-modifiers in a single NP.

We will consider the ordering of noun modifiers in pre- and post-position, and also both simultaneously. Our goal in this case is to avoid the implementation

[2] Lexical choice is often assumed to take place *before* surface realisation proper, as in [3]. However, our experiments are not committed to any particular NLG architecture.

[3] The choice for n-gram models of order 2 and 3 only was motivated by the amount of available training data.

of complex ordering constraints (or otherwise forcing the application to provide information in this level of detail) by using a n-gram model to select the appropriate configuration. This issue will be investigated in our Experiment 4.

Finally, the output sentences in a NLG system are of course expected to be grammatical, and a surface realisation module will normally implement a number of rules to enforce agreement and other linguistic constraints. Some of these constraints may however be difficult (or costly) to model unless a full grammar of each target language is considered. In [4], for instance, some long-distance dependencies are not fully supported by the surface realisation system, and the underlying application is expected to provide (or otherwise inherit from a predefined template) the correct gender and number attributes of certain sentence constituents.

We will address one particular kind of long-distance dependency that is not currently supported by the system described in [4], namely, the issue of verb and complement agreement. For instance, in 'The man who sold this house is very old', a morphologically-rich language such as Portuguese will require gender and number agreement between the subject (man) and the complement (old), which makes an interesting test case for the n-gram approach. We will consider those dependencies established in both active and passive voice, and our goal in both cases is once again to substitute language-dependent rules for a trainable statistical language model. These issues will be investigated in Experiment 5 and 6 described in the next section.

3 Evaluation

In this section we present the settings and results of Experiment 1-6 introduced in the previous section. Decisions made with the aid of language models of order 2 and 3 (called our Bigram and Trigram systems) will be compared to a number of baseline systems deemed relevant to each task. In all cases, the evaluation work will measure Edit-distance, NIST, BLEU and Accuracy scores obtained by each systems with respect to a manually-built reference set.

Briefly, Edit-distance is the traditional Levenshtein's distance (i.e., the number of insert, delete and substitute operations needed to make both strings identical.) Accuracy is taken to be the number of exact matches between the two strings, being assigned the value 1 if both strings are identical, or zero otherwise.[4] NIST [10] and BLEU [11] are widely-used evaluation metrics for Machine Translation systems based on n-gram counts[5].

3.1 Lexical Choice Preferences

The first experiment considers the lexical choice preferences of NP heads in subject position. To this end, 40 sentences were extracted from the corpus, making

[4] As an overly strict evaluation metric that expects exact string match, Accuracy scores are presented for illustration purposes only.

[5] BLEU scores range from 0 to 1, being 1 equal to 100% accuracy. NIST scores do not have an upper bound, but higher NIST scores stand for higher accuracy.

a Reference set. Next, multiple versions of each sentence were generated by making use of all possible surface forms for each NP head, taken from a thesaurus of Portuguese nouns. The thesaurus conveyed, on average, 3 alternatives for each NP found in the Reference data, and thus 120 alternative sentence realisations were produced out of the original 40-sentences set.

In order to select the most likely surface realisation form for each sentence, we consider the use of bigram and trigram language models compared against two baseline systems: a frequency-based approach that selects, for each concept, the most common word found in the corpus, and a strategy that simply chooses a random word out of each synset. The results are presented in Table 1. Recall that best results correspond to *lower* Edit-distance and *higher* Accuracy, NIST and BLEU scores.

Table 1. NP head lexical choice

System	Edit-distance	Accuracy	NIST	BLEU
Statistical 2-gram	2.88	0.60	9.048	0.955
Statistical 3-gram	1.73	0.75	9.151	0.969
Frequency-based	2.95	0.60	9.013	0.954
Random	4.70	0.40	8.842	0.928

Next, we consider the issue of lexical choice preferences as applied to VP head selection. The task in this case is considerably more complex than in the case of NP head selection given that (Portuguese) verbs tend to have a large number of synonyms: for instance, each verb synset in our data conveys an average of 10 synonyms.

Keeping all other sentence constituents unchanged, the 40 sentences in the Reference set used in Experiment 1 were re-generated, this time allowing only the main VP head constituents to vary according to the synonymous found in the thesaurus. As a result, 400 alternative sentences were produced. Following the same strategies from the previous experiment (Bigram, Trigram, Frequency-based and Random generation), the results were obtained as in Table 2.

Table 2. VP head lexical choice

System	Edit-distance	Accuracy	NIST	BLEU
Statistical 2-gram	4.98	0.13	8.731	0.863
Statistical 3-gram	3.88	0.30	8.877	0.893
Frequency-based	3.25	0.42	8.891	0.907
Random	5.38	0.13	8.659	0.852

Finally, in order to assess the limits of the n-gram approach to lexical choice, Experiment 1 and 2 were combined into a single task. In other words, the 40 Reference sentences were once again re-generated while attempting all possible

NP and VP surface realisation forms available from the thesaurus, producing a set of 1164 sentences in total. The results of the Bigram, Trigram, Frequency-based and Random strategies are presented in Table 3.

Table 3. Combined NP-VP head lexical choice

System	Edit-distance	Accuracy	NIST	BLEU
Statistical 2-gram	8.25	0.08	8.375	0.829
Statistical 3-gram	6.05	0.25	8.607	0.868
Frequency-based	6.08	0.20	8.552	0.867
Random	7.88	0.05	8.338	0.822

3.2 Ordering Constraints

In a second class of experiments we investigated whether strict linguistic constraints (and not merely domain preferences) that are notoriously difficult to capture in the form of grammar rules could be enforced by the use of statistical language models. To this end, we take the issue of ordering of noun modifiers as our working example.

A Reference set of 40 sentences was manually built as follows. Each sentence conveyed a simple subject-verb-complement structure in the undergraduate project domain. NP heads in subject position conveyed one or two modifiers each, and their correct position was always fixed, that is, cases in which more than one reading were possible were avoided.

The ordering of modifiers was evenly distributed across the data: in cases of single modifier, half instances required a pre-modifier (e.g., 'the red book') and half required a post-modifier (e.g., 'the book mentioned (in section 8)'[6]); in the case of two simultaneous modifiers, their instances were evenly distributed in three categories: using two pre-modifiers (e.g., 'the small red book'), using two post-modifiers[7], and using one pre-modifier and one post-modifier.

In our corpus, it is often the case that a noun is modified by prepositional phrases (PPs), as in, e.g., 'the cover of the book', and since PPs widen the distance between sentence constituents, these expressions pose a considerable challenge to the use of n-gram models in surface realisation. Thus, regardless of the number or position of modifiers, in this experiment we consider two kinds of NPs: those conveying a single noun head, and those conveying a noun head attached to a prepositional phrase. The number of sentences using a single NP or an NP-PP attachment was also evenly distributed in the Reference set.

Keeping all other constituents unchanged, the 40 sentences in the Reference set were generated considering all possible orderings of modifiers for the nouns in subject position. As a result, 132 alternative sentences were produced, with an average of 3.2 realisations for each Reference sentence.

[6] Examples of post-modifier usage may seem more natural in our target language (Portuguese) as in, e.g., 'O livro citado'.

[7] Once again, these are commonly seen in Portuguese, as in 'o prazo máximo permitido' ('the latest deadline allowed'.)

Unlike previous Experiment 1-3, which focused on domain preferences that could in principle be gauged at by means of corpus analysis, the present experiment addresses strict linguistic constraints that the text generator has to adhere to, making a comparison against a frequency-based approach unsuitable. For this reason, in Experiment 4 we will limit our analysis to the results of the n-gram models as compared to a single baseline system that selects one of the possible modifier orderings at random. The results are shown in Table 4.

Table 4. Ordering of noun modifiers

System	Edit-distance	Accuracy	NIST	BLEU
Statistical 2-gram	5.15	0.65	7.244	0.814
Statistical 3-gram	5.35	0.63	7.201	0.795
Random	9.73	0.27	6.831	0.559

3.3 Agreement Constraints

Finally, we examined the role of n-gram language models in verb and complement agreement, a problem which becomes obviously more complex, from a n-gram perspective, as additional words are inserted within the agreeing constituents.

Our starting point was the Reference set built for the previous Experiment 4 (on modifier ordering constraints). In order to increase the scope of the present investigation, we considered not only the 40 original sentences, but also modified versions of them in which the distance between verb and complement was increased by inserting one or two adverbs. Thus, a Reference set of 120 sentences was built, divided in three groups conveying 0, 1 or 2 intermediate constituents.

The complement gender (male/female) and number (singular/plural) attributes were evenly distributed across the data, although we did not expect an effect on this. The following is an example of each group.

(a) The students are keen to learn.
(b) The students are not keen to learn.
(c) The students are not very keen to learn.

We used 40 * 3 = 120 sentences in a standard subject + verb + complement order in active voice, making the Reference set for Experiment 5. In addition to that, we designed a separate Experiment 6 using a Reference set in which the same 120 instances were re-written in passive voice. By breaking the most typical n-gram chains in this way, we expect to have made the agreement task considerably more complex, allowing us to further assess the possible benefits and limitations of the n-gram-based approach.

In both Experiment 5 (agreement in active voice) and 6 (passive voice), we applied the same procedure adopted in the previous experiments. Keeping all other sentence constituents unchanged, our tests consisted once again of over-generating the Reference sentences, this time allowing only the gender and

number of the complement term to vary. In doing so, four alternative surface realisations for each sentence were generated, producing 480 output sentences in Experiment 5 and another 480 output sentences in Experiment 6. The results produced according to the Bigram, Trigram and Random strategies are summarised in Table 5 and 6.

Table 5. Verb-complement agreement constraints (active voice)

System	Edit-distance	Accuracy	NIST	BLEU
Statistical 2-gram	0.08	0.93	7.557	0.980
Statistical 3-gram	0.08	0.93	7.557	0.980
Random	0.99	0.25	6.467	0.791

Table 6. Verb-complement agreement constraints (passive voice)

System	Edit-distance	Accuracy	NIST	BLEU
Statistical 2-gram	0.45	0.67	7.132	0.886
Statistical 3-gram	0.42	0.68	7.167	0.892
Random	0.95	0.30	6.473	0.754

4 Discussion

Experiment 1-3 on lexical choice preferences showed overall positive results for NPs, VPs and NP-VP combined. The task seems however more easily accomplished in the case of NP head lexical choice than in the case of VPs. This was to some extent to be expected as verbs in our data tend to have a much larger number of synonyms than nouns.

Regarding the NP head lexical choice task, the Trigram strategy (and, to a lesser extent, also Bigram) was superior to the others. By contrast, in the VP head lexical choice task, the frequency-based approach was best of all.

As for the combination of NP-VP lexical choice, all systems showed a relatively low performance, suggesting that n-gram models do not provide a suitable distinction between complex, structurally-dependent phenomena. Although in this case the Trigram and Bigram models were still the best options, our results suggest that these issues should be addressed separately, or with the aid of a more powerful language model (e.g., taking syntactic information into account.)

Experiment 4 - on ordering constraints of noun modifiers - showed once again that the use of n-gram language models outperform the baseline system. In this case, the Bigram model was best of all, although the difference between Bigram and Trigram results was not statistically significant. Despite the presently higher Accuracy scores if compared to, e.g., those obtained in the previous experiments, we notice that establishing the ordering of NP modifiers was in some ways more difficult than performing lexical choice, as observed by the measured Edit-distance, NIST and BLEU scores.

Finally, regarding Experiment 5-6, we observe that the statistical models were both remarkably efficient in the verb-complement agreement task, with a small (but non-significant) advantage of Trigram over Bigram. More importantly, although not shown in the previous section, these results were found to be constant regardless of the distance between verb and complement[8], i.e., regardless of using zero, one or two intermediate words[9]. In addition to that, the experiments showed that the results for sentences in active voice were considerably superior to those obtained in passive voice, which may be explained by the fact that active voice sentences are much more frequent in the training data.

5 Conclusions

This paper presented a number of experiments to model typical surface realisation decisions with the aid of n-gram language models as a first step towards a simplified development and use of NLG resources. To this end, three tasks were considered: the lexical choice of NP and VP head constituents, the ordering of noun modifiers and verb-complement agreement.

Our preliminary results suggest that, as in [5], the use of n-gram statistics may indeed improve surface realisation of morphologically-rich languages such as Portuguese in a number of ways. In particular, the NP head lexical choice and verb-complement agreement tasks were successfully accomplished with the aid of language models, and that may indeed free the underlying application from the burden of providing this level of linguistic detail to the language generator.

By contrast, the VP head lexical choice was to some extent less successful. This may be partially explained both by the small size of the training data set and by the naivety of the language models under consideration.

As future work, we intend to make use of more sophisticated models that take into account not only n-gram statistics, but also morphological and structural properties of text such as those supported by factored language models [12].

Acknowledgments. The authors acknowledge support by FAPESP and CNPq.

References

1. Reiter, E.: An Architecture for Data-to-Text Systems. In: European Natural Language Generation Workshop (ENLG-2007), pp. 97–104 (2007)
2. McRoy, S., Channarukul, S., Ali, S.S.: An augmented template-based approach to text realization. Natural Language Engineering 9(4), 381–420 (2003)
3. Gatt, A., Reiter, E.: SimpleNLG: A realization engine for practical applications. In: European Natural Language Generation Workshop, ENLG 2009 (2009)

[8] There was actually a small increase in NIST/BLEU scores as longer sentences were considered, which is to be expected given the nature of these n-gram-based metrics.

[9] Although we are presently unable to validate this claim, we assume that a sufficiently robust language model of higher order may in principle be able to cope with longer dependency relations in a similar fashion.

4. Novais, E.M., Tadeu, T.D., Paraboni, I.: Text Generation for Brazilian Portuguese: the Surface Realization Task. In: NAACL-HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas, pp. 125–131 (2010)
5. Langkilde, I.: Forest-based statistical sentence generation. In: Proceedings of ANLP-NAACL 2000, pp. 170–177 (2000)
6. Reiter, E., Sripada, S.: Human Variation and Lexical Choice. Computational Linguistics 28(4) (2002)
7. Bangalore, S., Rambow, O.: Corpus-based lexical choice in natural language generation. In: 38th Meeting of the ACL, Hong Kong, pp. 464–471 (2000)
8. Malouf, R.: The order of prenominal adjectives in natural language generation. In: Proceedings of ACL 2000, Hong Kong (2000)
9. Mitchell, M.: Class-Based Ordering of Prenominal Modifiers. In: Proceedings of the 12th European Workshop on Natural Language Generation, Athens, pp. 50–57 (2009)
10. NIST: Automatic Evaluation of Machine Translation Quality using n-gram Co-occurrence Statistics (2002), `http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf`
11. Papineni, S., Roukos, T., Ward, W., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: ACL 2002, pp. 311–318 (2002)
12. Bilmes, J., Kirchhoff, K.: Factored Language Models and Generalized Parallel Backoff. In: Proceedings of HLT/NACCL, pp. 4–6 (2003)

Text-to-Text Surface Realisation Using Dependency-Tree Replacement

Eder Miranda de Novais, Thiago Dias Tadeu, and Ivandré Paraboni

School of Arts, Sciences and Humanities, University of São Paulo (USP / EACH)
Av. Arlindo Bettio, 1000 - São Paulo, Brazil
{eder.novais,thiagoo,ivandre}@usp.br

Abstract. Surface realisation - the task of producing word strings from non-linguistic input data - has been the focus of a great deal of research in the field of data-to-text Natural Language Generation (NLG). In this work we discuss an alternative approach to surface realisation, in which we borrow NLG techniques from the sister field of text-to-text generation to implement text generation based on examples in natural language. Our approach is suitable to simpler applications that are not linguistically-oriented by design, and which may be able to provide only minimal input knowledge to the NLG module.

Keywords: Natural Language Generation, Surface Realisation.

1 Introduction

Data-to-text Natural Language Generation (NLG) systems are intended to produce text (or speech) from non-linguistic input data. Systems of this kind often follow a pipelined architecture as in [7], comprising data interpretation, micro planning, macro planning and surface realisation tasks. In this work we focus on the latter, that is, the task of producing word strings from a given semantic specification provided as an input.

Computational approaches to surface realisation range from grammar-based systems to template-based generation - see for example [8]. In all cases, there is the question of how to provide the required input to the surface realisation module: on the one hand, grammar formalisms provide wide coverage and robustness, but may require highly detailed knowledge as an input, which often makes systems of this kind difficult to adapt to applications that are not linguistically-oriented by design. On the other hand, template-based surface realisation is generally thought of as making NLG technology more accessible by simplifying the input specification (and, to some extent, reducing its flexibility.) However, even template-based generation may require a considerably rich input.

Let us consider, for example, the case of YAG [4]. YAG is a template-based surface realisation system that may accept as an input either feature structures or propositional semantics. The following is an example of the first kind of input specification, representing a discourse subject (George) that performs an act

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 326–335, 2010.

(understand) on a particular object (a book), in which both subject and object happened to be realised as pronouns as "He understands it", taken from [4]:

```
((template clause
 (process "understand")
 (agent   ((template noun-phrase)
           (np-type PROPER)
           (head "George")
             (gender MASCULINE)
             (pronominal YES)))
 (object ((template noun-phrase)
          (head "book")
          (pronominal YES))) )
```

The above representation is clearly expressive and sufficiently flexible for a wide range of applications. However, we may still ask whether it is possible to generate sentences from less, or even *minimal* input knowledge so as to reduce the costs of adapting the application to the surface realisation module. One such situation may occur, for example, when we happen to know in advance the sentences that the application is most likely to generate (e.g., based on a corpus on the application domain.) If this is the case, then it may be possible to simply select an example sentence that *resembles* the desired output, and specify only the (possibly minor) changes that need to take place to produce the required text. For example, given an example sentence extracted from corpora as (a), we may request certain dependency-tree operations (e.g., the replacement of the subject and object sub trees) to arrive at a structurally-similar target sentence as in (b).

(a) [We] will start [a new project] tomorrow.
(b) [Our students] will start [school] tomorrow.

In this paper we discuss one such 'select-and-replace' approach to surface realisation, in which we borrow NLG techniques from the sister field of text-to-text generation (that is, the generation of new text by modifying an existing one, as seen in many NLP tasks such as machine translation, text summarization, etc.) to implement a surface realisation module based on examples that are specified directly in natural language, and using minimal additional knowledge. In other words, we will use text-to-text generation techniques as part of a standard data-to-text NLG architecture to produce new sentences based on similarity.

Our present work has been developed in the context of an ongoing Q&A project that is intended to model the most frequent exchanges between undergraduate students participating in a particular project and their professors. In this application, a corpus of target sentences describing a well-defined domain is available, and many of the existing sentences tend to resemble each other due to the re-currency of question and answer patterns. Put together, these are ideal conditions for investigating a select-and-replace approach to surface realisation.

2 Current Work

In order to develop our surface realisation module, we used a Brazilian Portuguese corpus of emails exchanged between students who are members of a particular undergraduate project and their professors. The section of the corpus conveying the professors' replies was segmented and subsequently parsed. From this dataset, 597 syntactically-structured templates in the undergraduate project domain were extracted.

Each template represents a corpus sentence with variable fields - or slots - to be filled in with values provided by the underlying application. In our current implementation we consider three kinds of variable fields: agent, action and patient constituents. Everything else in the sentence is simply canned text. Each variable field is assigned a dependency-tree structure representing a noun or verb phrase. These dependency-trees are, by default, inherited from the corpus, but they may be partially or totally overwritten by the application.

Our work focuses on the generation of word strings from these sentence structures, which can be viewed as a form of text-to-text generation. To this end, two methods for sentence generation are considered: (1) selecting an existing template and replacing some or all of its default constituents, or (2) creating a new sentence from a full input specification. In what follows we discuss each method (1,2) in turn, and then we address the related issue of synonymy.

2.1 Dependency-Tree Replacement

The first surface realisation method under consideration consists of producing a new sentence using a structurally-similar corpus sentence as a basis. In this case, the underlying application is expected to specify a target template id and, optionally, replace some or all of its constituents by providing new dependency-trees to occupy the desired slots.

The input to the surface realisation module in this case is a template id[1] and an arbitrary set of replacement values (e.g., gender, number, lemma, tense etc.) for individual sentence fields, which may include also entire structures (i.e., new dependency-trees) to replace the existing agent, patient and action constituents. Based on this input, the realisation algorithm performs three main steps: first, the sentence tree is traversed and all values that have been explicitly specified as part of the input are replaced; second, the main verb form is adjusted, if necessary, in order to agree with the subject (namely, when subject gender or number has changed); finally, the modified sentence is re-generated as a word string.

An example of this form of text-to-text surface realisation works as follows. At the most basic level, an application may simply select the required template to produce the desired output without any changes as in (c); with additional knowledge available, the application may also modify some of the sentence constituents, e.g., by setting the action value to [should finish] and the patient value

[1] Currently our work does not address the issue of template selection. This could be implemented, for instance, by using sentence similarity metrics as in [9].

to [work], producing the sentence in (d). This may ultimately be changed even further as in (e), in which case the agent value was set to [teacher] using a possessive determiner, the action value was set to [give] in future tense, and the patient value was set to [lecture] using an indefinite article, leaving only the sentence structure intact (besides the canned text component 'next week'.)

(c) [They] [will complete] [their task] next week.
(d) [They] [should finish] [their work] next week.
(e) [Our teacher] [will give] [a lecture] next week.

As in [2], we assume that the mappings from concepts to dependency-trees are domain-dependent, and that they are to be provided by the application. For testing purposes, we simply extracted all NPs and VPs from the corpus and created a forest of surface realisation forms representing the most typical domain concepts.

When a dependency-tree is replaced by a structure provided by the application, basic agreement operations (e.g., between subject and main verb) will take place to ensure grammaticality. In our work this task is accomplished with the aid of a Portuguese lexical database. The following is an example of sentence reorganisation in which the plural agent [All classes] in (f) is set to a singular concept [Your class] in (g), forcing the main verb to adjust in order to re-establish number agreement as in (h).

(f) [All classes] [terminate] at 1pm.
(g) [Your class] ∗[terminate] at 1pm.
(h) [Your class] [terminates] at 1pm.

2.2 Full-Sentence Specification

Besides using templates extracted from corpora as a basis to generate new sentences, we consider also a second surface realisation method: generating a sentence from scratch without any previous example. Although in principle defeating the idea of text-to-text generation, the full-sentence specification method is particularly useful in the generation of simpler sentence structures, in which case it may be more convenient[2] to specify a new sentence from scratch rather than selecting a template and commanding a large number of changes. Full-sentence specification is intended to be an additional facility to the select-and-replace method, and the combination of both is expected to provide us with a wide range of NLG tools for surface realisation based on minimal input knowledge.

In order to allow full-sentence specification, we defined an 'empty' template structure representing a standard <*agent action patient*> sentence and additional canned text in free position. In this case, the application is expected to provide the values of all variable fields, that is, none of the constituent values are inherited. Additional templates of this kind (e.g., representing a sentence in passive voice etc.) may be easily supported, but this currently requires new grammar rules definitions.

[2] That is, it may require less input knowledge.

The following is an example of full-sentence specification for the target "You are talking about the lecture on Statistics", presented here in simplified form (the actual representation would require a large number of additional parameters that are, in this example, left unspecified.) We notice that in this case there is no obvious advantage to our approach if compared to, e.g., the use of input feature structures in YAG [4] and, as previously mentioned, full-sentence specification is intended to be only complementary to the select-and-replace method.

Agent = [personal-pronoun, male, singular, 2pers, you]
Patient = [definite, male, singular, lecture, [on, [statistics]]
Action = [gerund, be, [talk, [about]]]

2.3 Synonymy

Regardless of generating a sentence from an existing example or building it from scratch, there is the question of how to determine the appropriate lexical choice to represent domain concepts. In a template-based approach this issue tends to be minimized by the fact that most words are pre-defined in the template structure, and no lexical choice as such needs to be performed. In the present case, for instance, one could simply define a large number of dependency-trees (which can be viewed as 'lower-order' templates for NPs and VPs) mapping every domain concept onto fixed content words. However, this may be overly strict: if we are to generate sentences from minimal input knowledge and with reasonable flexibility, then the choice of dependency-tree and content words should ideally be modelled as two independent tasks. This will require lexical choice to handle, at the very least, synonymy.

Each domain concept may have a large number of synonymous words that could in principle be used in different contexts, and choosing the most appropriate word string is not always straightforward. For example, suppose that the underlying application intends to refer to a domain concept 'teacher' by choosing a word from a synset containing terms such as 'tutor', 'instructor', 'professor', 'coach' and so on. In this case, in order to keep input specification as simple as possible, we would like to free the application from the burden of choosing the right wording for each situation.

The issue of synonymy could in principle be viewed as a matter of domain preference. This could be tackled, for example, by simply choosing the most *frequent* term in the domain. However, word choice is often context-dependent, and frequency alone is not sufficient to account for variation as seen in real language use. For instance, in our undergraduate project domain the words 'tutor' and 'professor' may be used interchangeably as a general term to denote the professor in charge of a tutorial. Nevertheless, in most cases there is a clear preference for one of the two words, and making the appropriate choice will require some level of contextual information to be taken into account.

Likewise [3] and others, we make use of n-gram language models as an aid to the lexical choice task. More specifically, we intend to over-generate all possible alternative realisations of a sentence, and select the most likely output according

to a language model of Brazilian Portuguese. The present use of n-gram statistics is however limited to the issue of synonymy, leaving other aspects of the lexical choice task (and which may also make use of statistical language models, as in the case of constituents agreement, for example) to be discussed elsewhere.

3 Evaluation

In order to evaluate our surface realisation approach we used the collection of templates on the undergraduate project domain as described in the previous section. Three kinds of test were considered: template-based generation using dependency-tree replacements, full-sentence specification, and statistically-driven lexical choice. In all cases we used a number of intrinsic evaluation metrics to compare the system output to a reference data set written by human judges (and which for this reason was kept relatively small.)

Regarding the dependency-tree replacement generation method, we developed four experiments to assess the system's ability to create new sentences from corpus examples as follows. First, 19 templates were randomly selected from the corpus, and four sets of replacement operations were applied over their agent and action constituents. In Experiment 1, the existing agent constituent of each sentence was replaced by a singular indefinite description; in Experiment 2, the agent constituents were replaced by a plural demonstrative description; analogously, Experiment 3 used a singular proper name, and Experiment 4 used a plural definite description. In all four experiments the existing action constituents were replaced by two different actions, making 19 templates * 4 agents * 2 actions = 152 test sentences in total.

The test sentences were created manually by performing the required replacements to each corpus sentence, making four *reference* sets. Subsequently, the same 152 sentences were also generated automatically, making four *system* sets. By comparing each reference-system pair, we would like to show that the system performs constantly well regardless of the kind of replacement operation being requested. However, we notice that since the templates were randomly selected, the requested operations were not always entirely adequate and, as a result, not all (system and reference alike) sentences were semantically well-formed. For that reason, we do not expect total agreement between the two sets, a point that we shall come back to in the next section.

Each reference and system sentence pair was compared by measuring Levenshtein's Edit-distance (that is, the number of insertion, deletion and substitution operations required to make both strings identical), Accuracy (the number of exact matches between the two strings[3]), BLEU and NIST scores, both of which widely-used in the evaluation of Machine Translation systems[4].

[3] Accuracy is assigned the value 1 if both strings are identical, or zero otherwise, and in this sense is a very strict evaluation metric, presented here for illustration purposes only.

[4] BLEU scores range from 0 to 1, being 1 equivalent to 100% accuracy. NIST scores do not have an upper bound, but higher NIST scores stand for higher accuracy.

Results for Experiments 1-4 are summarized in Table 1. Since the experiments use different reference sets - and hence are not directly comparable to each other - BLEU and NIST scores presented below are only intended to show that the system performance was stable across all four experiments.

Table 1. Dependency-tree replacement results

#	Experiment	Edit-distance	Accuracy	BLEU	NIST
1	Singular indefinite	5.47	0.316	0.781	7.277
2	Plural demonstrative	5.53	0.211	0.770	7.254
3	Singular proper name	5.53	0.263	0.781	7.253
4	Plural definite	5.53	0.211	0.770	7.256

Next, in order to test the system's ability to generate text from a full-sentence specification, that is, without any corpus template as a basis, Experiment 5 was developed as follows. Leaving all lexical choice issues aside (i.e., choosing content words, verb tenses etc. in advance), we considered all possible sentences that could be produced in the standard <*agent action patient*> linear order from a set of six predefined agent concepts (e.g., student, teacher etc.), six action concepts (develop, must-finish etc.) and three patient concepts (project, work and internship.)

The resulting 6 agents * 6 actions * 3 patients = 108 sentences were once again produced both manually by a human judge[5], and also automatically by the system. In this case, we would like to show that the system produces well-formed sentences given a full input specification conveying agent, action and patient constituents.

Results for Experiment 5 are as follows. Unlike Experiment 1-4, in Experiment 5 the selected constituent values always yielded to meaningful sentences and, as a result, the human annotator did not have to strive to produce them. Quite expectedly, in all cases (100%) the system output was identical to the manually-produced sentences.

Finally, likewise [3], we also examined the role of statistical language models applied to the lexical choice task, focusing on the issue of synonymy only. In Experiment 6, a collection of 40 templates was randomly selected from the corpus and each template was filled in with six different action constituents (the same constituents used in previous Experiment 5), making 40 templates * 6 actions = 240 sentence specifications.

Keeping all the surrounding text exactly as seen in the corpus, once again we manually composed these 240 sentences making a reference set. Next, the system was applied to the generation of the same 240 sentences while varying the wording of NPs. To avoid combinatory explosion, in this experiment we considered only the variation of nouns in agent position. The synonyms were taken from a thesaurus of Brazilian Portuguese in which each agent concept may

[5] As a highly precise, well-defined task, in this case it was not necessary to make use of additional judges to ensure agreement.

have from 1 to 9 alternative realisations, with an average of 3 instances each. As a result, 720 sentences were generated from the original 240 input specifications.

In order to select the most appropriate word string for each input, two lexicalisation strategies were considered: selecting the most likely output based on a bigram statistical language model, and the same using a trigram model. These strategies were compared to two baseline methods: selecting the most frequent surface realisation form found in the corpus, or simply selecting a random alternative from each synset. The language models, when applicable, were built from a 40-million words collection of newspapers articles in Brazilian Portuguese[6].

Each strategy produced a distinct system set comprising 240 instances of output sentences. By comparing each system set to the (manually-built) reference set, we would like to show that statistically-driven lexical choice outperforms both frequency-based and random lexical choice. As in the previous experiments, this evaluation task was carried out by measuring Edit-distance, Accuracy, BLEU and NIST scores. Results are presented in Table 2.

Table 2. Synonymy results

System	Edit-distance	Accuracy	BLEU	NIST
2-gram filter	2.22	0.717	0.957	9.781
3-gram filter	2.42	0.683	0.953	9.739
Frequency-based	3.05	0.600	0.939	9.563
Random	3.62	0.508	0.927	9.439

4 Discussion

The use of intrinsic evaluation metrics described in the evaluation work required us to make use of manually-built reference data, which limited evaluation to a few hundred sentences in each experiment. While this is clearly insufficient for large-scale evaluation, we consider results of Experiment 1-6 to be satisfactory as a first assessment of the present work.

Regarding the dependency-tree replacement task addressed in Experiment 1-4, the results are considerably below what could be expected for an otherwise straightforward procedure. Although not shown in previous Table 1, those results were actually influenced by a small number of extreme cases: in one set, for example, there was an edit-distance value of 27, which is unacceptably high for a task that should ideally achieve zero edit-distance.

As pointed out in the previous section, situations in which the system sentence differed greatly from the reference sentence were mainly due to our (with hindsight misguided) assumption that dependency-tree replacement operations could be assessed using *any* sentence, regardless of being semantically well-formed or not. However, as the selected operations were not always applicable to the test templates, in some cases it was not possible to judge adequately how or where the

[6] Our domain corpus alone, conveying only 597 sentences, was not sufficiently large to build a robust language model of this kind.

required replacement should take place. As a result, human annotators made in some cases arbitrary decisions that, although intuitive, could not be matched by the system. This was mainly the case when a more complex template structure (e.g., with subordinate or coordinate clauses) was involved.

For example, given the template sentence in (i), and commanding the (inappropriate) replacement of the action constituent to [will break], human annotators may, perhaps intuitively, obtain (j), whereas the system, following the information provided by the parser, may obtain (k). In either case, it is not clear what it means to replace an action constituent in this way and, if so, whether the system evaluation should actually be penalized.

(i) If it is just a matter of time, then he should finish his homework.
(j) If it is just a matter of time, then he ∗[will break] his homework.
(k) If it ∗[will break] just a matter of time, then he should finish his homework.

To sum-up, we believe that the present results still suffice to demonstrate that dependency-tree replacements perform satisfactory, but we do not dispute that practical, real examples of language use would have provided us with a clearer picture for Experiment 1-4.

In Experiment 5, by contrast, the full-sentence specification task showed 100% accuracy, even though including an additional operation (i.e., the insertion of patient constituents), making the task in some ways more complex than in Experiment 1-4. As mentioned in the previous section, those positive results were indeed to be expected as all the requested operations yielded to semantically well-formed sentences that human annotators did not have difficulties to compose, and which could be accurately reproduced by applying standard grammar rules.

Finally, Experiment 6 successfully confirmed that synonymy may indeed be guided by statistical language models, as suggested in previous work in the field. The statistical filters (and, particularly in our case, the use of a simple bigram model) produced results that are considerably superior to both frequency-based and random choice, and this is so despite the fact that we used a relatively small amount of training data.

5 Final Remarks

In this paper we have described an approach to surface realisation which is partially based on text-to-text generation techniques. Our current system is able to generate sentences from minimal input specification by performing dependency-tree replacement operations over example sentences, and it is also able to produce new sentences from scratch (i.e., from a full input specification.) In addition to that, the choice between synonymous words was successfully implemented with the aid of n-gram language models.

The three tasks - generation by dependency-tree replacement, generation from full-sentence specification and lexical choice - were evaluated in a series of experiments based on intrinsic evaluation metrics. Preliminary results suggest that

our approach is suitable to NLG applications that are not linguistically-oriented by design, and which may be able to provide only minimal input knowledge to the NLG module. This is particularly the case of applications in which (likewise our underlying Q&A project) a corpus of target sentences is available, and in which many of the existing sentences tend to resemble each other due to the re-currency of question and answer patterns.

As future work we will extend the system coverage and apply language models to other lexicalisation tasks, e.g., in order to help establish constituent agreement, an issue that is particularly relevant to highly-inflected languages such as Portuguese. Moreover, we intend to adapt the existing surface realisation module to a real Q&A application under development in order to run a large-scale, extrinsic evaluation exercise as well.

Acknowledgments. The authors acknowledge support by FAPESP and CNPq.

References

1. Bangalore, S., Rambow, O.: Corpus-based lexical choice in natural language generation. In: 38th Meeting of the ACL, Hong Kong, pp. 464–471 (2000)
2. Gatt, A., Reiter, E.: SimpleNLG: A realization engine for practical applications. In: European Natural Language Generation Workshop, ENLG 2009 (2009)
3. Langkilde, I.: Forest-based statistical sentence generation. In: ANLP-NAACL 2000, pp. 170–177 (2000)
4. McRoy, S., Channarukul, S., Ali, S.S.: An augmented template-based approach to text realization. Natural Language Engineering 9(4), 381–420 (2003)
5. Oh, A., Rudnicky, A.: Stochastic language generation for spoken dialogue systems. In: ANLP-NAACL 2000 Workshop on Conversational Systems, pp. 27–32 (2000)
6. Pereira, D.B., Paraboni, I.: Statistical Surface Realisation of Portuguese Referring Expressions. In: Nordström, B., Ranta, A. (eds.) GoTAL 2008. LNCS (LNAI), vol. 5221, pp. 383–392. Springer, Heidelberg (2008)
7. Reiter, E.: An Architecture for Data-to-Text Systems. In: European Natural Language Generation Workshop (ENLG 2007), pp. 97–104 (2007)
8. van Deemter, K., Krahmer, E., Theune, M.: Real versus template-based NLG: a false opposition? Computational Linguistics 31(1) (2005)
9. Seno, E.R.M.: Um método para a fusão automática de sentencas similares em português. PhD Thesis, University of São Paulo (2010)

A Comparative Analysis of Centering-Based Algorithms for Pronoun Resolution in Portuguese

Fernando José Vieira da Silva[1], Ariadne Maria Brito Rizzoni Carvalho[1], and Norton Trevisan Roman[2]

[1] Institute of Computing, State University of Campinas, Campinas, Brazil
ra085324@students.ic.unicamp.br, ariadne@ic.unicamp.br
[2] School of Arts, Sciences and Humanities – University of São Paulo, São Paulo, Brazil
norton@usp.br

Abstract. Pronominal anaphora resolution consists in finding a referent for a given pronoun. Although being essential for many natural language processing systems, such as automatic translators, text generators and summarizers, there is a myriad of issues regarding this task, particularly when there is more than one possible referent for a given pronoun. Over the years, several approaches have been proposed to deal with this challenge, usually taking only syntactic information into account. On the other hand, methods based on Centering Theory rely on concepts such as coherence, for instance, to do the job. In this work, we describe our implementation and evaluation of existing centering-based algorithms for pronominal resolution in Portuguese. As a result, we indicate both the strong and weak points of each of the tested algorithms, thereby helping other researchers to make a more informed decision about which method to use.

1 Introduction

Anaphora is a linguistic phenomenon that happens when there is an abbreviated reference to another element in the text. This abbreviation is called an *anaphor*, while the element to which it refers is called a *referent*. Although, in this work we focus only on pronominal anaphora, i.e., when the anaphor is a pronoun, the challenge of finding a referent for a given anaphor still remains a great one, raising considerably in the presence of more than one possible referent, as in the sentence: "O **cachorro** de João fugiu esta manhã. **Ele** latiu e assustou o carteiro." ("João's **dog** fled this morning. **It/He**[1] barked and scared the postman."). In this sentence, both "cachorro"(dog) and "João" are candidates for the anaphor "Ele", since both agree in gender and number with the pronoun, being this kind of ambiguity solved at the semantic level only.

On the search for solutions to automatic pronominal anaphora resolution, several strategies have been explored. The approaches studied in this work are

[1] In Portuguese, both "it" and "he" are translated to "ele".

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 336–345, 2010.

based on Centering Theory[8], which, among other things, deals with the notion of local discourse coherence, i.e., coherence between utterances [9], as determined by the way the information is presented in the text. Therefore, Centering Theory provides some insights to the pronoun resolution task, by always looking for a referent that would keep the discourse's coherence, there still existing, when in the absence of such a referent, the intuition that such discourse segment is not coherent.

In this work, we describe the results obtained on implementing and analyzing such algorithms by running them over a small corpus of texts adapted from the papers in which the algorithms are presented, along with other papers that deal with Centering Theory (*e.g.* [5,3]). By doing so, we intend to provide other researchers with information that may help them in the choice for the most suitable algorithm, according to the properties of the analyzed corpus. We also discuss about each method's main features, describing not only their precision, but also their computational complexity. Based on our results, we also make some suggestions for modifications in these algorithms.

The remaining of this paper is organized as follows: the next section presents the Centering Theory and its concepts for pronominal anaphora resolution; Section 3 shows a conceptual algorithm, purely based on the ideas introduced by the theory, as described in [8] and [3], designed to serve as a basis for comparison to the other algorithms. The implementation of the BFP, S-List and LRC algorithms are described in Sections 4, 5 and 6, respectively. In Section 7, we discuss some characteristics and behaviour of these algorithms and finally, Section 8 shows our conclusions and final comments.

2 Centering Theory

Centering Theory is a system of rules and constraints that govern the relations between the text goals and the author's linguistic choices to express the flow of ideas [1]. Based on Grosz and Sidner's Theory [7], Centering Theory introduces the notion of local coherence, i.e., coherence between utterances in the same discourse segment. Discourse coherence is then measured by the number of times the writer changed the subject in the text, depending on the sequence the information is presented.

According to Centering Theory, the participants' focus of attention can be captured by some "centers", which exist in every utterance of the discourse. Let U_n be the *n-th* discourse utterance, and U_{n-1} its precedent; each utterance U_n has the following associated elements [1,7,8]:

- ***Cf*(U_n) (Forward Looking Center):** It is composed of every entity which may be the focus on the next utterance, consisting of all pronouns and noun phrases found in the current utterance. The elements in the $Cf(U_n)$ are ordered according to their syntactic function in the text, with the following preference: subject > object > others;
- ***Cp*(U_n) (Preferred Center):** It is the most probable element to be the center in the next utterance. The Preferred Center is the best ranked element in $Cf(U_n)$;

– ***Cb*(U_n) (Backward Looking Center):** It corresponds to the best ranked element from the previous utterance ($Cf(U_{n-1})$) which is realized as a pronoun in the current utterance.

In order to measure the text coherence, as indicated by the change of focus through utterances, Centering Theory specifies all center transitions according to the change or maintenance of entities that compose the $Cf(U_n)$, $Cp(U_n)$ and $Cb(U_n)$ of each utterance. The transitions considered in this work, including the Shifting-1 transition introduced in [3], are:

1. **Center Continuation:** This transition indicates that the information about the same entity is presented consecutively, without any other entity in the discourse; i.e., the focus of attention is maintained. Formally, it occurs when: $Cb(U_n) = Cb(U_{n-1})$ and $Cb(U_n) = Cp(U_n)$;
2. **Center Retaining:** Although there exists a reference to another entity in the discourse, its focus is kept the same, thereby continuing the same theme. It occurs when $Cb(U_n) \neq Cp(U_n)$ and $Cb(U_n) = Cb(U_{n-1})$;
3. **Shifting-1:** The discourse focus is slightly shifted, changing the text theme. It is expected, however, that this new theme will be maintained in the next utterance. It happens when $Cb(U_n) \neq Cb(U_{n-1})$ and $Cb(U_n) = Cp(U_n)$;
4. **Center Shifting:** It indicates a change in the discourse focus. It occurs when $Cb(U_n) \neq Cb(U_{n-1})$ and $Cb(U_n) \neq Cp(U_n)$.

Based on the identified center transition in a specific discourse segment, one can determine its coherence, by considering a segment with Center Continuation more coherent than another with Center Retaining which, in turn, is more coherent than one with Shifting-1. Finally, segments with Center Shifting are considered the less coherent among all. The intuition behind this rule – which in fact has some experimental support (*e. g.* [3]) – is that a text that shows information about the same entity in a sequential way is more coherent than another that mentions several entities in a interleaved way.

Finally, along with defining a measure for discourse coherence, Centering Theory also presents some rules regarding pronoun realization, to wit, if some element of $Cf(U_{n-1})$ is realized by a pronoun in $Cf(U_n)$, then this element must be the $Cb(U_n)$. When the utterance has several pronouns, one of them must be the $Cb(U_n)$. If, however, there is only one, then it must be the $Cb(U_n)$.

3 Conceptual Algorithm

Since Centering Theory was not specifically designed for pronominal resolution, it does not deal with some related problems, such as finding referents for intrasentential anaphora, for example. Due to this fact, algorithms like BFP, S-List and LRC need some customizations in order to handle these issues, or even to become more accurate. In order to evaluate these customizations, we have compared these three algorithms with a fourth one, named Conceptual Algorithm, built exclusively on the concepts introduced by Centering Theory, and which should be used as a benchmark.

1. Eu não vejo João há alguns dias.
(I haven't seen João for a few days).
Cf = {Eu, João, dias}
Cb = {}
Cp = {Eu}

2. Carlos acha que ele está estudando para suas provas.
(Carlos thinks he is studying to take tests).
Cf = {Carlos, Ele = João, provas}
Cb = {João}
Cp = {Carlos}
Transition = CENTER SHIFTING

3. Eu acho que ele foi para o interior com Linda.
(I think he went to the countryside with Linda).

(1st Set : ele=Carlos)
Cf = {Eu, ele, interior, Linda}
Cb = Carlos
Transition = CENTER SHIFTING

(2nd Set - THE CHOOSEN ONE! : ele=ele=João)
Cf = {Eu, ele, interior, Linda }
Cb = ele
Transition = CENTER RETAINING

Fig. 1. Discourse sample and sets built by Conceptual Algorithm

In a nutshell, the algorithm builds a universal *Cf* set for each utterance. This set contains every entity mentioned in the utterance, grammatically ordered according to the following preference [3]: subject > object > object 2 > others > adjunct. Then, for each pronoun in the utterance, the algorithm builds all possible <*Cf*, *Cb*> sets, where *Cb* is some entity from $Cf(U_{n-1})$ which agrees in gender and number with the pronoun under consideration. In this implementation, a different set is built for each entity in $Cf(U_{n-1})$ that agrees with this pronoun.

The algorithm then chooses one of the <*Cf*, *Cb*> sets for each pronoun it finds in the current utterance, by analyzing the center transitions for each set and choosing the most coherent one. If, however, there is more than one set with the same center transition, being this transition the most coherent, the algorithm chooses the one whose *Cb* is better ranked in $Cf(U_{n-1})$. This referent is then taken for the *Cb* of the chosen set. As such, the algorithm does not try to obtain a single <*Cf*, *Cb*> set for each utterance, but instead, it considers one different set for each pronoun in that utterance, being the corresponding *Cb* its referent.

When the algorithm chooses a set for some pronoun in the utterance, it seeks the referent for that specific pronoun only. Thus, even if there are other pronouns

in the utterance, they are still mentioned in the *Cf* of the set but with no indication of their referents[2]. Therefore, the *Cb* of this set which, in turn, is the referent for the pronoun, is the only entity realized by a pronoun in the *Cf* of the set. Naturally, the *Cb* is the best ranked entity from $Cf(U_{n-1})$ that is realized in the *Cf*, which is in line with Centering Theory's rules.

Figure 1 illustrates the process of finding referents for pronouns in an utterance. In this figure, the third utterance exemplifies the variation of sets for the pronoun "ele" (he) and the algorithm's choice according to the center transition. In the example, two sets are created for the pronoun. The second one is chosen because it presents a Center Retaining transition between utterances 2 and 3, which is more coherent than the Center Shifting that would happen if the first set were chosen. So the algorithm concludes that the pronoun "ele" refers to "João". As for the algorithm's computational complexity, the task of finding referents for all pronouns of a given utterance is dominated by the step in which the algorithm chooses a set for every pronoun in that utterance, taking $O(n^3)$ time, where n is the number of entities realized in that utterance.

4 The BFP Algorithm – Brennan, Friedman e Pollard

The algorithm introduced by Brennan, Friedman and Pollard [3] was the first and possibly the best well-known algorithm based on concepts from Centering Theory [6], comprising the following steps:

1. Creation of all possible <*Cf*, *Cb*> sets for the utterance;
2. It filters the sets created in step 1;
3. It chooses a set with the highest coherence, as determined by Centering Theory.

In the first step, all entities in U_n are grammatically ordered, according to the following preference: subject > object > object 2 > others > adjunct. Then all possible *Cf* elements are created as follows: for each pronoun in the utterance, an expanded set is generated, composed by elements in the format <*en*, *ref*>, where *en* is the pronoun under consideration and *ref* is a candidate for referent. This expanded set contains one element for each entity in $Cf(U_{n-1})$ that agrees in gender and number with the pronoun. Next, each possible *Cf* is generated, containing one element from each different expanded set and each noun phrase in the utterance. Finally, each possible *Cf* is combined with each *Cb* candidate, where every entity in $Cf(U_{n-1})$ and the empty element are possible *Cb*s. This combination results in a list of $O(nm^2)$ possible <*Cf*, *Cb*> sets, where n is the number of entities realized in the current utterance, and m is the number of entities realized in the previous utterance. The algorithm then filters and discards some sets from that list, according to the rules described below:

1. ***Cb* not realized:** Every set whose *Cb* is not realized by some entity in that set's *Cf* is discarded;

[2] The referents for these pronouns are found separately by the algorithm.

2. ***Cb* is not the best ranked element in *Cf*(U_{n-1}):** If the *Cb* element is realized by some entity in the *Cf*, but there is another entity which is better ranked in *Cf*(U_{n-1}) being also realized by a pronoun, then this set is discarded;
3. **Filter by contra-indexes:** If there are two contra-indexed[3] pronouns in the *Cf*, referring to the same entity, or if there is a pronoun that refers to an entity with which it is contra-indexed, then this set is discarded.

After these filters are applied, the algorithm chooses a <*Cf*, *Cb*> set for the utterance, by taking the center transition for each possible remaining set and choosing the most coherent one. The algorithm's computational complexity is dominated by the filtering step, which takes the time $O(n^5)$.

5 The S-List Algorithm

This algorithm is characterized by the use of only one data structure to store the entities of the discourse – the S-List, which is composed by every entity realized in a specific utterance, ordered as described below:

1. **Information Status** – The Information Status of an entity indicates if it was recently introduced into the context or if it has been mentioned earlier. According to this criterion, entities are classified as OLD, MED or NEW, where the preference order for entities is OLD > MED > NEW. The classes that compose these groups are the following [12,13]:
   - **OLD:** Composed by entities classified as EVOKED or UNUSED (whose meaning will be made clearer later on);
   - **MED:** Formed by entities classified as INFERRABLE, CONTAINING INFERRABLE and ANCHORED BRAND NEW. It is assumed that CONTAINING INFERRABLE and ANCHORED BRAND NEW are equivalent in practice;
   - **NEW:** Only the BRAND NEW class belongs to this group;
2. **Sentence position in the discourse – where the entity is realized**: Entities closer to the end of the discourse are better ranked than the others;
3. **Position within the utterance**: Entities closer to the beginning of the utterance are better ranked than the others.

The algorithm then adds each entity to the S-List, respecting the ordering rules mentioned above. After processing an utterance, it discards the unrealized entities. The search for a referent then becomes a simple lookup through the S-List. The entities's classification according to their Information Status is as follows [16][4]:

[3] In this implementation, according to Binding Constraints defined by Chomsky [10], two pronouns or noun phrases are considered contra-indexed if both cannot refer to the same entity.

[4] In [16], however, neither INFERRABLE or CONTAINING INFERRABLE are considered.

- **EVOKED:** If the entity is realized by a pronoun;
- **BRAND NEW:** If the entity is preceded by an indefinite article;
- **UNUSED:** If the entity is not preceded by a determiner[5]. Only proper names fall into this category;
- **ANCHORED BRAND NEW:** If none of the conditions above hold;

The S-List algorithm, implemented as described in [16], has $O(n^2)$ computational complexity for finding referents for all pronouns in a given utterance.

6 LRC Algorithm – *Left-Right Centering*

The LRC Algorithm, proposed in [15], appends every entity in the utterance to a partial *Cf*. When it finds a pronoun, the algorithm starts looking intra-sententially for a referent in this partial *Cf* from left to right. The first found entity that agrees in gender and number with the pronoun and, in addition, respects the Binding Constraints defined by Chomsky [10], is taken as referent. At the end, the pronoun is inserted in the partial *Cf*. If no suitable referent is found, then the algorithm looks for a referent in $Cf(U_{n-1})$, starting with the best ranked entity, also reading it from left to right. When the process is finished, the $Cf(U_n)$ is built by simply ordering the partial *Cf* grammatically, as in the BFP algorithm. This algorithm's computational complexity for finding the referent for every pronoun in a given utterance is $O(n^2)$.

7 Discussion

Since it is based only on Centering Theory's rules, the Conceptual Algorithm has no other strategy for choosing $<Cf, Cb>$ sets than measuring center transition coherence. This fact leaves room for the possibility that, for an utterance having two pronouns that agree in gender and number with each other, the algorithm may choose the same referent for both, leading to incoherent interpretations in some cases. Along the same lines, there is also the possibility that the algorithm may assign the same referent to two contra-indexed pronouns [10]. The BFP algorithm, on the other hand, deals with contra-indexed pronouns through its filtering process.

Nevertheless, a common criticism to the BFP Algorithm concerns its computational complexity [15]. Overall, the responsibility for the exponential complexity can be mainly attributed to the $<Cf, Cb>$ set generation, since this step creates $O(nm^2)$ sets, which must be considered by all forthcoming steps, either at the filtering or final set identification stage. Another drawback present in both BFP and Conceptual algorithms is their incapability to solve intra-sentential anaphora, while S-List and LRC may solve them correctly.

Concerning intra-sentential anaphora, unlike the S-List Algorithm, LRC always prefers intra-sentential referents to inter-sentential ones, which might be

[5] Determiners are definite and indefinite articles and pronouns tagged as determiners by the *PALAVRAS* [2] parser.

harmful to its success rate, when analyzing a text containing mostly inter-sentential anaphora. On the other hand, its ordering strategy allows the S-List Algorithm to choose an inter-sentential referent instead of a possible intra-sentential one, depending on its Information Status. Such a mechanism may improve its accuracy over an heterogeneous corpus, with a balanced number of intra- and inter-sentential anaphora. The ability for solving intra-sentential anaphora, however, does not necessarily means an advantage for the algorithms S-List and LRC, since in texts where inter-sentential anaphora are majority, BFP and Conceptual algorithms may have advantage.

In this work, we also run these algorithms to find referents for personal pronouns over the Summ-it corpus [4], which is composed of articles extracted from the science section of a well known Brazilian newspaper, containing 118 personal pronouns in total, with 7% of intra-sentential anaphora. The articles are segmented in text spans used to define RST[6] relations [14] and we consider these spans as utterances. In our implementation, only nouns and personal pronouns are available to become part of Cf, Cb or S-List, where each noun is considered as a realization of a different entity and each personal pronoun is considered as a realization of a previous mentioned entity. For this evaluation however, we haven't implemented an algorithm for checking Binding Constraints, so none of the algorithms check them. The results are shown in Table 1. In this table, "Intra Result" and "Inter Result" columns give, respectively, the percentage of success in finding referents for intra-sentential and inter-sentential anaphora[7], while the column "Overall Result" presents the percentage of success in finding referents for any kind of anaphor. The "Not Found" column indicates the percentage of anaphora to which the algorithm could not find any referent.

Table 1. Results obtained from running Conceptual, BFP, S-List and LRC algorithms over the Summ-it corpus

Algorithm	Complexity	Intra Result	Inter Result	Not Found	Overall Result
Conceptual	$O(n^3)$	0%	37%	31%	34%
BFP	$O(n^5)$	0%	37%	31%	34%
S-List	$O(n^2)$	78%	34%	25%	37%
LRC	$O(n^2)$	78%	34%	25%	42%

As expected, both Conceptual Algorithm and BFP did not solve any intra-sentential anaphora. On the other hand, S-List and LRC have shown a much better success rate for solving intra-sentential anaphora than for inter-sentential ones. However, inter-sentential results for the two first algorithms are better than for the last two. Still, overall results for S-List and LRC are better than for the algorithms that do not solve intra-sentential anaphora, even for this corpus where inter-sentential anaphora are majority. In this corpus, the highest success rate is reached by the S-List algorithm, whose advantage in comparison

[6] Rhetorical Structure Theory.

[7] The success rate is relative to the number of intra- and inter-sentential anaphora.

to LRC may be explained by the high frequency of inter-sentential anaphora in the corpus, since LRC always prefers intra-sentential referents. As we did not check for binding constraints, the equality between Conceptual and BFP results is comprehensible.

Finally, none of the algorithms is supposed to find referents that are not realized in the previous utterance, which, in turn, is a limitation of the Centering Theory, when used for anaphora resolution. We believe that it may be the main reason for the cases where the algorithms were not able to find a referent.

8 Conclusion

In this work we presented four algorithms based on Centering Theory for automatic pronominal anaphora resolution, describing their implementations. We also analyzed their main features, along with some possible situations where each of them would succeed or not. Results show that, overall, the best suited algorithm for a corpus with a balanced number of intra-sentential and inter-sentential anaphora is S-List, due to its ability to solve intra-sentential anaphora without discarting inter-sentential referent candidates. Even in our experiments, where we used a corpus with a much higher number of inter-sentential anaphora, S-List still kept the best success rate.

Another interesting result from our experiment is the high number of anaphora that the algorithms were not able to resolve. A deeper study of these cases might lead to thorough modifications on algorithms based on Centering Theory for anaphora resolution, by having algorithms that consider, as referent candidates, entities realized in utterances other than the one immediately before the anaphor.

Other avenues for future work include the evaluation of some variations of the originals S-List and LRC algorithms. For example, when two entities in an utterance have the same Information Status, S-List could take the entity which is closer to the pronoun, instead of preferring the one closer to the beginning of the utterance. As for LRC, we might try to implement the intra-sentential search in the opposite way, starting from right to left. The reason to believe these approaches could result in improvements for the algorithms is the fact that Hobb's Algorithm [11] presents such a preference, and experiments made in [15] found a better success rate in a corpus with more than 71% of intra-sentential anaphora. Finally, the S-List algorithm makes no distinction between two entities with different Information Status if they belong to the same group. Take, for example, the UNUSED and EVOKED classes. Currently, entities belonging to these classes have the same weight in S-List, being part of the OLD group. However, it could be the case that it is more probable that an EVOKED entity be realized again in the text than an UNUSED entity be realized for the first time, in which case it might be worth distinguishing one from the other. Practical experiments on distinguishing entities with different Information Status in the same group would be interesting.

References

1. Aires, A., Coelho, J., Collovini, S., Quaresma, P., Vieira, R.: Avaliação de centering em resolução pronominal da língua portuguesa. In: 5th International Workshop on Linguistically Interpreted Corpora of the Iberamia 2004, Puebla, Mexico, pp. 1–8 (November 2004)
2. Bick, E.: The Parsing System "Palavras". In: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. University of Arhus, Arhus (2000)
3. Brennan, S.E., Friedman, M.W., Pollard, C.: A centering approach to pronouns. In: ACL 1987, pp. 155–162. ACL, Stanford (1987)
4. Carbonel, T.I., Collovini, S.S., Coelho, J.C., Fuchs, J.T., Rino, L.H.M., Vieira, R.: Summ-it: Um corpus anotado com informações discursivas visando à sumarização automática. In: Proceedings of XXVII Congresso da SBC: V Workshop em Tecnologia da Informação e da Linguagem Humana? TIL, Rio de Janeiro, Brazil, pp. 1605–1614 (2007)
5. Cristea, D.: Discourse theories and technologies. In: Tutorial at ICON 2004, Hyderabad, India (2004)
6. Grosz, B.J., Joshi, A.K., Weinstein, S.: Towards a computational theory of discourse interpretation (1986)
7. Grosz, B.J., Sidner, C.: Attention, intentions, and the structure of discourse. Computational Linguistics 12(3) (1986)
8. Grosz, B.J., Weinstein, S., Joshi, A.K.: Centering: A framework for modeling the local coherence of discourse. Computational Linguistics 21, 203–225 (1995)
9. Poesio, M., Eugenio, B.D., Stevenson, R., Hitzeman, J.: Centering: A parametric theory and its instantiations. Computational Linguistics 30, 309–363 (2004)
10. Santorini, B., Kroch, A.: The syntax of natural language: An online introduction using the Trees program (2000), `http://www.ling.upenn.edu/~beatrice/syntax-textbook`
11. Santos, D.N.A., Carvalho, A.M.B.R.: Hobbs' algorithm for pronoun resolution in portuguese. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 966–974. Springer, Heidelberg (2007)
12. Strube, M.: Never look back: an alternative to centering. In: Proceedings of the 17th International Conference on Computational Linguistics, pp. 1251–1257. Association for Computational Linguistics, Morristown (1998)
13. Strube, M., Hahn, U.: Functional centering – grounding referential coherence in information structure. Computational Linguistics 25, 309–344 (1999)
14. Taboada, M., Mann, W.C.: Introduction to rst (rhetorical structure theory), `http://www.sfu.ca/rst/01intro/intro.html`
15. Tetreault, J.R.: Analysis of syntax-based pronoun resolution methods. In: ACL 1999, pp. 602–605. College Park, MD (1999)
16. Tetreault, J.R.: A corpus-based evaluation of centering and pronoun resolution. Computational Linguistics 27, 507–520 (2001)

Specification and Evaluation of a Spanish Conversational System Using Dialogue Models

Ivan V. Meza, Lisset Salinas, Esther Venegas, Hayde Castellanos, Alejandra Chavarría, and Luis A. Pineda*

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), Universidad Nacional Autnoma de México
{ivanvladmir,liz}@turing.iimas.unam.mx,
{esthervenegas,alejandra3101}@hotmail.com,
hcvcastellanos@gmail.com, luis@leibniz.iimas.unam.mx

Abstract. In this paper, we present a new specification, implementation and evaluation of the "guess the card" system. This is a conversational system with spoken Spanish and vision capabilities that plays a game with members of the general public in a permanent stand at a science museum. The system has been built using a methodology and programming environment based on the notion of dialogue model specification and interpretation, that we have been developing over the years. The present system uses the latest version of the formalism and improves considerably over previous versions [8]. In particular, the present version takes advantage of the discourse structure, increases modularity, and incorporates general and flexible recovery strategies. An evaluation of the system is also presented.

Keywords: Spoken Dialogue System, Dialogue Models, Multimodal Systems.

1 Introduction

"Guess the card" is a Spanish spoken dialogue system that handles multimodal input and output including the interpretations of images through computer vision and the display of pictures and animations to support linguistic behavior. The system is a permanent stand at a Universum science museum of the National Autonomous University of Mexico (UNAM) [8]. The game is played by 10-to-14-year-old children mostly, who have the goal of guessing a card chosen by the system randomly; for this they are allowed to ask a number of questions (up to four) about the features of the images on the cards.

The system's architecture is centered around the notion of dialogue models specification and interpretation. A dialogue model is a representation of a

* We thank the enthusiastic support of the members of the DIME and Golem group at IIMAS and at Science Museum Universum-UNAM. We also gratefully thank the support of grants CONACYT 81965 and PAPPIT-UNAM IN-104408 and IN-115710.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 346–355, 2010.

conversational protocol which is defined in advance through analysis. These protocols relate potential speech acts and expectations in conversational situations with the actions that the system needs to perform when such speech acts are performed by the interlocutor or an expectation is meet in the environment, also at the situation. Instances of dialogue models are assembled dynamically during the interaction producing rich and natural conversations. The dialogue manager (DM) interprets the dialogue models continuously along the conversation. This is a domain independent program interpreter that has been used to build a number of different applications. For instance, [2,1,13].

The main tenant of our approach is that dialogue acts are expressed and interpreted in relation to a conversational context that is shared between the participants. We pose that the interpretation context has two main parts: a global context that holds for whole of the conversational domain, and a specific context that is built dynamically along each particular conversation. Dialogue models can be thought of as representations of the global context, and the history of a conversation as the specific context. In this paper, we define a new specification which modularizes the dialogue models and simplifies the system's construction, testing and maintenance. Second, we propose mechanisms to deal with the specific context of the dialogues to improve the interaction with the system, for instance by remembering the name of user to addressing him or her during the conversation. Finally, we have added recovery strategies which permit to deal better with recognition and interpretation errors during the interaction.

This paper is organized as follows. Section 2 describes the task. Section 3 introduces the framework for the specification and interpretation of dialogue models. Section 4 presents the dialogue models for the current application. In this section the behavior of the system is also illustrated with a full actual human-user conversation. The evaluation of the present version of the system is presented in Section 5. Section 6 is devoted to some implementation issues. Finally, in section 7 we present our conclusions.

2 The Task

The goal of the user is to deduce the card chosen by the system from a set of ten cards. For this, the user can ask up to four questions in spoken Spanish. The ages of the users are between 10 and 14 years old. The cards have astronomical motives (e.g., the sun, a telescope) and are located on a table in front of the user. In a typical session the system introduces itself, asks for the name and age of the child. Next, if required, the system explains the rules of the game. Then, there is the questions session properly. At the end of the interrogatory, the child is asked to place the card that he or she thinks is the right one in front of the camera; then, the system confirms whether the child has won the game, or tells the child which card was the right one. During the game the system responses are rendered using synthesized language, sound effects and images displayed on the screen.

3 Architecture and Dialogue Models

"Guess the card" system is based on an interaction oriented cognitive architecture for multimodal Human-Computer Interaction (HCI) [11]. This architecture has three conceptual levels that we call recognition-rendering, interpretation-specification and representation-inference, in addition to the semantic and episodic memory. The architecture is shown in Figure 4 at [11]. The recognition and interpretation modules correspond to the perceptual modalities. For the present application we use speech and visual perceptual processes. The goal of these modules is to assign an interpretation to input spoken and visual messages. On the other hand, the specification and rendering modules correspond to behavioral response. In this application we use speech and visual actions.

The representation-inference layer has the purpose of identifying the adequate action given an interpretation. This layer corresponds to the dialogue models that are used to specify practical dialogues. These represent conversational protocols in which the user "visits" a sequence of conversational *situations* with highly structured expectations about what can be expressed by the interlocutor or about the visual events that can occur in the world, which we call expected intentions or *expectations*. A dialogue model is represented as a directed graph (cycles are permitted). Situations are represented as nodes and edges are labeled with expectation and action pairs. If the expectation of an edge matches the current interpretation, then the corresponding action is performed. Situations can have one or more input and output expectation-action pairs. Situations are typed according to the modality of the expected input; the main types are *listening* (or linguistic) and *seeing* (or visual). There is also a special type of situation in which a full dialogue model is embedded called *recursive*. When a recursive situation is reached, the current dialogue model is pushed down into a stack, and the embedded model is interpreted, so the conversation as a whole has a stack structure. All dialogue models have one or more final situations, and when these are reached, the model's interpretation process is terminated. If there is a dialogue at the top of the stack, it is popped up and its interpretation is resumed; otherwise the dialogue is terminated. Dialogue models are represented through recursive transition networks, augmented with functions standing for expectations, actions and next situations (F-RTN) [10]. For further detail on the dialogue models specification consult [11].

Recognition is a bottom-up process that translates the input information into an image or pattern in a predefined coding format. No meaning or interpretation is assigned to the product at this recognition level. We refer to this kind of patterns as "uninterpreted images". For "guess the card" we use an Automatic Speech Recognizer (ASR) to translate speech into text and the computing vision algorithm SURF to codify the cards in terms of a set of salient points invariant to translation and rotation. The interpretation is a heterarquic process that given the *expectations* of the situation (top-down) and an uninterpreted images (bottom-up) assigns an interpretation to the input message. In the present application, each expected intention has an associated regular expression, which codifies a large number of forms that can be used to express such intention. So,

such regular expressions are applied to the text recovered by the ASR system, and whenever such a match holds, the concept associated to the regular expression is selected. Visual concepts are recovered by a similar process in which the SURF representation of the external image is matched with the SURF representations associated to the visual concepts that are available to the system. These perceptual associations are stored beforehand in the system's episodic memory. On the other hand, the propositional knowledge about the properties of the cards is stored in the system's semantic memory.

Output actions associated to expectations are specified through Multimodal Rhetorical Structures (MRS); these are lists of basic acts, defined along the lines of the Rhetorical Structure Theory (RST) [6].

4 Dialogues Models

For the present application, we identified six sub-dialogues which capture the structure of the task. These sub-dialogues are: name recognition (n) and age verification (a) for asking the user's name and verifying his or her age; instructions (i), for explaining the user the purpose and rules of the game; interrogation (I), for permitting the user to ask questions about the features of the cards; visualization (V), for permitting the system to see and verify the card selected by the user; and farewell (F), for thanking and telling goodbye to the user. In order to capture this structure we defined eight dialogue models. One for each of these sub-dialogues plus the greeting (G) and the main (M) dialogue models. The main dialogue model coordinates the greeting (G), the interrogatory (I), the visualization (V) and the farewell (F) sub-dialogues. The greeting consists of a greeting by the system plus the name recognition, age verification and instructions sub-dialogues, and the choosing of card by the system. Table 1 presents a full dialogue between the system and a child using this structure. The table shows the turns, the dialogue model to which they belong, the utterances produced by the user and the corresponding recognition hypotheses produced by the ASR system, the interpretation hypotheses, and the visual interpretations and display actions performed by the system.

A typical interaction begins at the starting situation of the main dialogue model. Figure 1 illustrates the main (M) dialogue model,with its initial situation S. As explained above, edges are labelled with the expectations and actions pairs, and they are directed to the following situation. In this situation, the only edge is labelled with the pair ϵ:*display_button*. Accordingly, the edge is transversed, the corresponding buttons are displayed on the screen, and the situation (C_I) is reached. The displayed buttons allow the user to start the game, to play an explanatory video, and to end the application. These three possibilities constitute the current system expectations, and whenever the user selects and option, the corresponding action and situation transition are performed by the system. Figure 1 also illustrates four recursive situations: greeting (G), interrogatory (I), visualization (V) and farewell (F). Whenever one of these situations is reached, the corresponding dialogue model is loaded and executed. In addition, when

Table 1. Example of interaction between the "guess the card" system and a user (Spk, Speaker; DM, Dialogue model; Inter., Interpretation)

Turn	Spk	DM	Spanish Utterance/`Recognition hypothesis`/ *English Translation*	Inter.	Visual actions
1	SYS	*M*	Hola a todos me llamo Golem y éste es el juego de adivina la carta, tenemos jugador, ¿Cúal es tu nombre?/*Hello everyone, my name is Golem and this is the "guess the card" game, we have a player, what is your name?*		Welcome image
2	USR	*n*	Octavio/`one`/*Octavio*	*notU*	
3	SYS	*n*	Estoy medio sordo,¿Cúal es tu nombre?/*I'm a little bit deaf, which is your name?*		
4	USR	*n*	Octavio/`hector luz`/*Octavio*	*name*(*hector*)	
5	SYS	*n*	Hola Hector, me da gusto jugar contigo/*Hello Hector, I'm glad to play with you*		
6	SYS	*a*	¿Me puedes decir tu edad?/*Could you tell me your age?*		
7	USR	*a*	Trece años/`trece años`/*Thirteen years old*	*age*(13)	
8	SYS	*a*	¿Seguro qué tienes trece aos?/*Are you sure you are thirteen years old*		
9	USR	*a*	Sí/`si`/Yes	*ok*	
10	SYS	*a*	Muy bien, entonces vamos a jugar/*Very good, let's play*		
11	SYS	*i*	¿Quieres que te explique las reglas del juego?/*Do you want me to explain the rules of the game?*		
12	USR	*i*	Sí/`si`/*Yes*	*ok*	
13	SYS	*i*	Presta atención ... Después de la ronda de preguntas compararemos mi carta y la que tú crees que elegí. Veremos si adivinaste. ¿Entendiste las instrucciones?/*Pay attention ... After that we will compare my card with the one you think I have chosen. We will see if you have guessed. Did you understand the instructions?*		A slideshow with examples
14	USR	*i*	Sí/`si`/*Yes*	*ok*	
15	SYS	*i*	Okay, fíjate muy bien en las cartas mientras escojo una/*Okay, look at the cards while I chose one.*		
16	SYS	*I*	Empecemos el juego, por favor hazme tu primera pregunta/*Let's start the game, please make your first question*		
17	USR	*I*	¿Es roja?/`aldo`/Is it red?	*notU*	
18	SYS	*I*	Me puedes preguntar otra característica que veas por favor/*Could you ask me about any other feature you see in the cards please ?*		
19	USR	*I*	¿Se parece un plato?/`parece un plato`/Does it look like a dish?	*feat*(*dish*)	
20	SYS	*I*	Lo siento, eso no está en mi carta. Has la siguiente pregunta/*Sorry, that is not on my card. Come on, ask your next question*		Image with a dish on it
21	USR	*I*	¿Es redonda?/`onda`/*Is it round*	*feat*(*wave*)	
22	SYS	*I*	No hay nada como una onda. Hazme la siguiente pregunta.*There is nothing like a wave. Make your next question*		Image with water waves
23	USR	*I*	¿Es azul?/`luz`/*is it blue?*	*feat*(*light*)	
24	SYS	*I*	No atinaste, en mi carta nada tiene luz propia. Okay, ésta es la última pregunta que me puedes hacer, píensala bien. *Sorry, there is nothing with its own light. This is your last question, think it very well*		An image with light
25	USR	*I*	¿Es rojo?/`es rojo`/Is ti red?	*feat*(*red*)	
26	SYS	*I*	Veo algo parecido, pero no es rojo/*I see something similar but not red.*		Image with red color
27	SYS	*V*	Por favor muestrame la carta/*Please show me the card*		
28	USR	*V*	**User shows card with the sun on it to the camera**	*card*(*sun*)	
29	SYS	*V*	No tengo muy buena vista. Es ésta la carta que me mostraste/*I don't have a good sight, is this the card that you showed to me?*		Image of the recognized card
30	USR	*V*	Sí/`si`/Yes	*ok*	
31	SYS	*V*	Lástima, la carta que había escogido es la del planeta mercurio/*Sorry, I had chosen the Mercury card*		Image of the Mercury card
32	SYS	*F*	Oh! Pero antes de irte por favor mira a la cámara. Gracias por jugar conmigo, regresa pronto/*Oh! But before you leave look at the camera. Thanks for playing with me, come back soon*		Camera takes a picture and shows it

the execution of the recursive dialogue model ends, it produces an action as a whole, which becomes the expectation that is met by one of the edges leaving the recursive situation. In this model, there are two possible outcomes in each of the recursive situations: the game continues (*continue*) or ends (*end*). This will depend on what happens inside the dialogue models, if users interactions are being interpreted properly, then the conversation is continued (outcome *continue*). However, if the system is not interacting with the user, the game is ended (outcome *end*). Outcomes can carry parameters, for instance at the situation G the sub-dialogue returns the card chosen by the system, and this is represented by the parameter C; also, the dialogue returns the chosen card (C) and the card mentioned by the user before his or her four chances (Y) at the situation I.

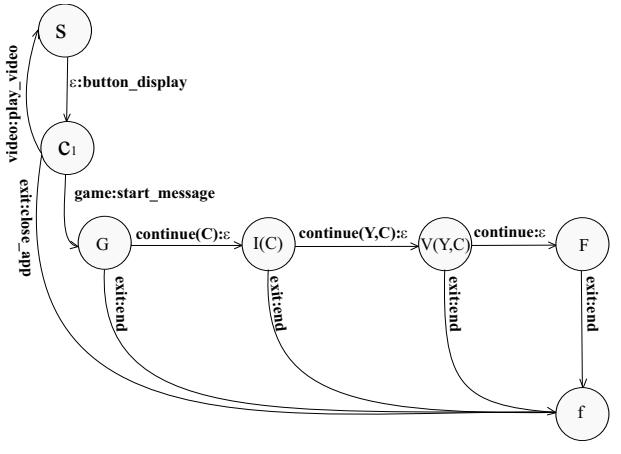


Fig. 1. Main dialogue model (M)

The interrogatory dialogue model is illustrated in Figure 2. This is a more complex dialogue model, as will be explained. This model starts at situation S which is reached and traversed automatically as explained below. Here the system asks the user to make his or her first question and reaches situation s_1. At this situation the system has the expectation that the user ask a question about a feature of the card. This is represented by $feature(X)$. $f(X)$ stands for an acknowledgment related to the feature X. Here, f is a function that is evaluated dynamically and produces the relevant rhetorical act as its value. At this point the situation n is reached, and a message is rendered indicating that there are a number of chances left, if any (*question_number*). The following situation depends on whether the user has exhausted his or her chances. This is also achieved by an explicit function. Functions standing for next situations are represented by a small dot at the end of the edge, and the values for all possible parameters are represented by circles reached by dashed lines starting at the dot representing the function. When there are still questions left the dialogue cycles over situation s_1; otherwise, the dialogue moves into the final situation e. At situation s_1 there is also the possibility that the user tries to guess the

card. This is represented in the dialogue model by the $card(Y)$ expectation. In this case, a message is rendered asking the user to confirm ($confirmation$) if he or she is indeed trying to guess the card and the situation s_2 is reached. Here, there are two expectations, whether the user confirms or denies this intention (ok or no). If the answer is yes the system will confirm that the listened card is the one the user is trying to guess. At this point the situation s_3 is reached. If the user wishes to guess the card, the dialogue model ends and the situation e is reached. The parameters (i.e. the chosen and the guessed card) are returned to the main dialogue ($conitnue(Y, C)$). This message is passed to the visual (V) dialogue model, where the test of whether the user guessed the card right is performed.

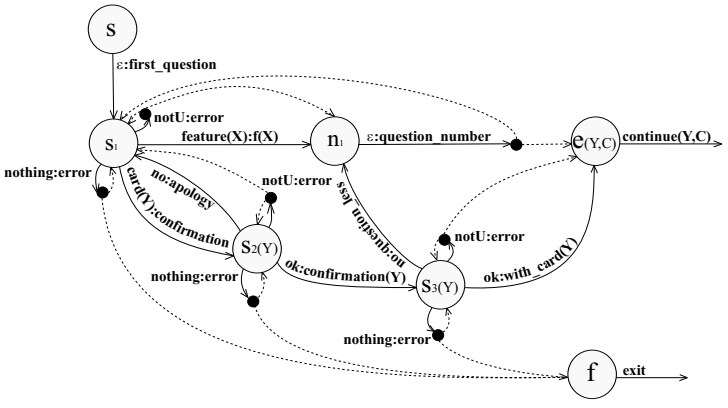


Fig. 2. Interrogation dialogue model (I)

We have also included two recovery strategies for speech communication. The first is used when the ASR module produces not output (i.e. there is nothing to listen to). The second is used when the system does not "understand" what the user says. Figure 2 illustrates these cases with the expectations: $nothing$ and $notU$. These are default expectations invoked when there are no other expectation met. The number of times a recovery model is invoked is also controlled through a function, making the dialogue more fluent and coherent. A misunderstanding that is not critical can be skipped over.

We have also improved the dialogue structure. The new version of the system takes advantage of the specific context of the conversation too. For instance, it remembers the name of the user to address him or her later on in the conversation (e.g. at the farewell). In addition, we have included confirmation questions at points of the conversation where conflict may arise. For instance, the questions in the interrogatory dialogue model (i.e., situation s_2 and s_3) where an early attempt to guess the card requires the user's and system's agreement. We also added a confirmation question for the age verification, so the conversation can proceed only if the child is old enough.

5 Evaluation

Evaluation is essential for designing, developing and optimizing spoken dialogue systems. It measures the performance of individual system components, dialogue strategies and the overall system performance. We have performed a preliminary evaluation of our system using the user-satisfaction questionnaire from the PARADISE framework [14]. For this, we asked ten children to play the game. All the children finished the game in 9 and half minutes in average. Three of the children guessed the right card. There were 15.5 user turns per dialogue. A previous version of the system reported 33.6, so the present specification has a considerable improvement. We attribute this to the addition of more robust recovery strategies. The visual system was able to identify the card at a rate of 1.2 tries per card. Table 2 summarizes the results obtained in the user satisfaction questionnaire. We found that the game by itself is hard for children, as it can be seen in the *task ease* factor. However, children felt confident about what to ask (*user expertise*). The quality of the ASR system needs to be improved considerably, since the children perceived that they are not being understood. Despite these shortcomings, all of the children wanted to play with the system again.

Table 2. Percentage of the "yes" answers to the user-satisfaction questionnaire

Factor	Percentage
TTS Performance	90%
ASR Performance	50%
Task ease	30%
Interaction pace	90%
User expertise	70%
System response	50%
Expected behavior	60%
Future use	100%

6 Implementation

We used the Open Agent Architecture framework (OAA) for communication between the recognition, interpretation, inference, representation-inference, specification and rendering modules [7]. The dialogue manager is implemented in *Prolog*. The ASR system is Sphinx3 [5]. We also developed a speech recognizer for children. For this, we collected the Corpus DIMEx100 Children. This is a corpus based on the Corpus DIMEx100 Adults [12]. The speech interpreter uses a word spotting technique based on regular expressions. For visual perception we used feature extraction and matching based on the Speeded-Up Robust Features (*SURF*) algorithm [3]. The SURF implementation is based on OpenCV [4] with a naive nearest neighbor search.

7 Conclusions

In this paper, a new specification, implementation and evaluation of the "guess the card" system was presented. This is a conversational system with spoken Spanish and vision capabilities that plays a game with members of the general public in a permanent stand at a science museum. In this application, children between 10 and 14 years old try to guess a card chosen randomly by the system. In order to do this, the users ask up to four questions about the features of the card. The architecture of the system is based on a cognitive architecture for multimodal Human-Computer Interaction (HCI).

This version of the system presents major improvements over previous ones. First, the task (and the conversation) is better structured, with eight dialogue models. This partition makes the specification, programming and debugging tasks much simpler. Second, we proposed mechanisms to deal with the specific context of the dialogues to improve the interaction with the system, such as by remembering the name of user in the farewell. We also added confirmation questions to assure agreement in sensitive parts of the conversation. Finally, we added recovery strategies which deal with speech errors and miss-interpretations.

The present application has been preliminary evaluated using the PARADISE framework. The quality of ASR system is still an issue and it needs to be improved considerably for the construction of robust applications. In this line, we are exploring the automatic generations of domain specific language models. We found that most children are able to complete the game, and are willing to play it again, showing a reasonable degree of user satisfaction. Communication failures are lower than in previous versions. This translates into fewer user turns and the shortening of the dialogues and a better user satisfaction [14].

References

1. Avilés, H., Alvarado-González, M., Venegas, E., Rascón, C., Meza, I., Pineda, L.A.: Development of a Tour–Guide Robot Using Dialogue Models and a Cognitive Architecture. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 512–521. Springer, Heidelberg (2010)
2. Avilés, H., Meza, I., Aguilar, W., Pineda, L.A.: Integrating Pointing Gestures into a Spanish-spoken Dialog System for Conversational Service Robots. In: Second International Conference on Agents and Artificial Intelligence (2010)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded-Up Robust Features. Computer Vision and Image Understanding 110(3), 346–359 (2008)
4. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library ORilley (2008)
5. Huerta, J.M., Chen, S.J., Stern, R.M.: The 1998 Carnegie Mellon University Sphinx-3 spanish broadcast news transcription system. In: Proc. of the DARPA Broadcast News Transcription and Understanding Workshop (1999)
6. Mann, W.C., Thompson, S.: Rhetorical structure theory: Towards a functional theory of text organization. Text 8(3), 243–281 (1998)
7. Cheyer, A., Martin, D.: The Open Agent Architecture. Journal of Autonomous Agents and Multi-Agent Systems 4(1), 143–148 (2001)

8. Meza, I., Pérez-Pavón, P., Salinas, L., Aviles, H., Pineda, L.A.: A Multimodal Dialogue System for Playing the Game Guess the card. Procesamiento de Lenguaje Natural 44, 131–138 (2010)
9. Pineda, L.A., Estrada, V., Coria, S., Allen, J.: The Obligations and Common Ground Structure of Practical Dialogues. Revista Iberoamericana de Inteligencia Artificial 11(36), 9–17 (2007)
10. Pineda, L.A.: Specification and Interpretation of Multimodal Dialogue Models for Human-Robot Interaction. In: Sidorov, G. (ed.) Artificial Intelligence for Humans: Service Robots and Social Modeling, Mexico, pp. 33–50 (2008)
11. Pineda, L.A., Meza, I., Salinas, L.: Dialogue Model Specification and Interpretation for Intelligent Multimodal HCI. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 20–29. Springer, Heidelberg (2010)
12. Pineda, L.A., Castellanos, H., Cuétara, J., Galescu, L., Juárez, J., Llisterri, J., Pérez, P., Villaseñor, L.: The corpus DIMEx100: Transcription and evaluation. Language Resources and Evaluation (2009)
13. Rascón, C., Avilés, H., Pineda, L.A.: Robotic Orientation towards Speaker for Human-Robot Interaction. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 10–19. Springer, Heidelberg (2010)
14. Walker Marilyn, A., Litman, D.J., Kamm, C.A., Kamm, A.A., y Abella, A.: Paradise: A framework for Evaluating Spoken Dialogue Agents, pp. 271–280 (1997)

A Half-Way Semantics toward Collaborative Behavior in Interagent Dialogues*

M. Julieta Marcos, Marcelo A. Falappa, and Guillermo R. Simari

National Council of Scientific and Technical Research (CONICET)
Artificial Intelligence Research & Development Laboratory (LIDIA)
Universidad Nacional del Sur (UNS), Bahía Blanca, Argentina
{mjm,mfalappa,grs}@cs.uns.edu.ar

Abstract. In this paper we extend an existent approach to formalizing agents' behavior in collaborative dialogues. We build on an existent abstract dialogue framework, for which some collaborative semantics have already been defined, and contribute by developing a new semantics which covers the in-betweens of two of the previous, namely the *basic collaborative* and the *full collaborative* semantics. The new intermediate semantics captures the behavior of agents who make an effort toward achieving the highest degree of collaboration (full collaborative semantics) although they do not always succeed. We provide a complete example of the proposed semantics using *Propositional Logic Programming with Negation as Failure*.

1 Introduction

Agents in a multi-agent system need to communicate for different reasons: to resolve differences of opinion or conflicts of interests, to cooperate for solving problems or finding proofs, or simply to communicate each other about pertinent facts [1]. This gives rise to a variety of dialogue types, such as the ones identified by Walton and Krabbe [2]: *persuasion*, *inquiry*, *negotiation*, among others. Some dialogue types, such as the *inquiry* dialogue, are *collaborative* in the sense that the agents are willing to share any relevant knowledge to the topic at issue, having no other ambition than achieving the right conclusion on the basis of all the information they have.

Much work has been done to develop formal models of these dialogues, both in the collaborative and non-collaborative categories (*e.g.*, [3–9]). A key notion which has received special attention in various works (*e.g.*, [10–15]) is that of *relevance of moves*. In that sense, Marcos *et al.* have defined in [15] an abstract dialogue framework which provides an environment for studying the collaborative behavior of agents in terms of an abstract relevance notion, for a simplified model of interaction where the only possible move is to make a *contribution* (that is, to expose a subset of one's private knowledge).

The authors in [15] tackle the *distributed relevance problem*, which refers to situations in which the relevant knowledge at certain step is distributed among different sources in such a way that none of the parts is by itself relevant, leading to a kind of

* This research is partially supported by Sec. Gral. de Ciencia y Tecnología de la Univ. Nac. del Sur (PGI 24/ZN18) and CONICET (PIP 112-200801-02798).

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 356–365, 2010.

deadlock configuration, since none of the participants can move so as to make the dialogue progress. They define a set of *collaborative semantics*, each characterizing a different degree of cooperation, which is directly related to a particular handling of the distributed relevance problem. The *utopian collaborative* semantics describes an idealistic, in most cases impractical behavior of collaborative agents. The *basic collaborative* semantics portrays the simplest practical approach which consists merely in disregarding the existence of distributed contributions. The *full collaborative* semantics, lastly, corresponds to a more complex practical approach which manages to guarantee that relevant contributions will always come to light, even when they are distributed.

In this paper we attempt to extend the work done in [15] by developing a new semantics which covers the in-betweens of the *basic collaborative* and the *full collaborative* semantics just mentioned. This half-way semantics captures the behavior of agents who make an effort toward achieving the highest degree of collaboration, although they do not always succeed. The formalization of this intermediate behavior may be useful both for categorizing existent approaches, and for developing new ones in cases where full collaboration is too hard to be put into practice, and not strictly required.

The rest of this work is organized as follows. Sections 2 and 3 summarize respectively the dialogue framework and semantics proposed in [15]. Section 4 introduces a new semantics, namely the *advanced collaborative semantics*, and provides illustrative examples within a dialogue framework instance taken from [15], which uses *Propositional Logic Programming with Negation as Failure* for knowledge representation and reasoning. Finally, Section 5 highlights the main contributions of this work, as well as pointing out some issues left for future work.

2 Abstract Dialogue Frameworks

This section and the next one summarize the main concepts and results presented in [15], so the reader is referred to that work for further detail. The authors assume the following languages to be involved in a dialogue: the *Knowledge Representation Language* $\mathcal{L}$ for expressing the information exchanged by the agents, the *Topic Language* $\mathcal{L}_T$ for expressing the topic that gives rise to the dialogue, the *Outcome Language* $\mathcal{L}_O$ for expressing the final conclusion (or *outcome*), and a language $\mathcal{L}_I$ for agent identifiers.

Agents are represented as pairs $\langle \mathrm{id}, \mathrm{K} \rangle$, noted K_{id}, consisting of a private finite knowledge base $\mathrm{K} \subseteq \mathcal{L}$ and an agent identifier $\mathrm{id} \in \mathcal{L}_I$. A *dialogue* is defined as a tuple $\langle \mathrm{t}, \langle \mathrm{m}_j \rangle, \mathrm{o} \rangle$ where $\mathrm{t} \in \mathcal{L}_T$ is the dialogue topic, $\langle \mathrm{m}_j \rangle$ is a sequence of *moves*, and $\mathrm{o} \in \mathcal{L}_O$ is the dialogue outcome. A *move* is a pair $\langle \mathrm{id}, \mathrm{X} \rangle$ where $\mathrm{id} \in \mathcal{L}_I$ is the identifier of the *speaker*, and $\mathrm{X} \subseteq \mathcal{L}$ is her *contribution*. Note there is only one type of move: to publish a set of knowledge. Besides, only dialogues in which agents' contributions are subsets of their private knowledge (called *admissible dialogues*) are considered.

An *abstract dialogue framework* is then defined as a tuple $\mathfrak{F} = \langle \mathcal{L}, \mathcal{L}_T, \mathcal{L}_O, \mathcal{L}_I, \mathcal{R}, \Phi, \mathrm{Ag} \rangle$, also briefly noted $\langle \mathcal{R}, \Phi, \mathrm{Ag} \rangle$, where Ag is a finite set of *agents*, $\mathcal{R} \subseteq 2^{\mathcal{L}} \times 2^{\mathcal{L}} \times \mathcal{L}_T$ is a *relevance notion*, and $\Phi : 2^{\mathcal{L}} \times \mathcal{L}_T \Rightarrow \mathcal{L}_O$ is a *reasoning model*. A *relevance notion* represents a criterion for determining, given certain already known information and a topic, whether it would be relevant to add certain other information (*i.e.,* to make a contribution). If $(\mathrm{X}, \mathrm{S}, \mathrm{t}) \in \mathcal{R}$, then X is said to be a

t-relevant contribution to S *under* $\mathcal{R}$, and it is noted $X\mathcal{R}_t S$. A *reasoning model*, as usual, is understood as a mechanism for obtaining a conclusion about a topic, on the basis of an individual knowledge base.

The following partially instantiated dialogue frameworks are defined in [15] (and will be used later in this work) using *Propositional Logic Programming* (*PLP*) and its extension with *Negation as Failure* (*PLP*$_{naf}$) as the underlying formalism for knowledge representation and reasoning. It is assumed that the reader is familiarized with the concept of derivation in *PLP* (noted $\vdash$) and *PLP*$_{naf}$ (noted $\vdash_{naf}$)[1].

- $\mathfrak{F}^{lp} = \langle \mathcal{L}^{lp}, \mathcal{L}_{Facts}, \{\text{Yes}, \text{No}\}, \mathcal{L}_I, \mathcal{R}_t, \Phi^{lp}, \text{Ag}\rangle$ where $\mathcal{L}^{lp}$ is the set of rules and facts in *PLP*, $\mathcal{L}_{Facts} \subset \mathcal{L}^{lp}$ is the subset of facts (which in this case works as the Topic Language) and $\Phi^{lp}(S, h) = \text{Yes}$ if $S \vdash h$, and No otherwise.
- $\mathfrak{F}^{naf} = \langle \mathcal{L}^{naf}, \mathcal{L}_{Facts}, \{\text{Yes}, \text{No}\}, \mathcal{L}_I, \mathcal{R}_t, \Phi^{naf}, \text{Ag}\rangle$ where $\mathcal{L}^{naf}$ is the set of rules and facts in *PLP*$_{naf}$ and $\Phi^{naf}(S, h) = \text{Yes}$ if $S \vdash_{naf} h$, and No otherwise.

The authors also introduce an intuitive relevance notion, called *natural relevance notion* and noted $\mathcal{N}_t^{\Phi}$, which can be defined for any given reasoning model Φ as follows:

$$X\ \mathcal{N}_t^{\Phi} S \text{ if, and only if, } \Phi(S, t) \neq \Phi(S \cup X, t).$$

That is, a contribution is considered relevant if, and only if, its addition alters the conclusion achieved by reasoning. Those dialogue frameworks in which the relevance notion is the natural one associated to the reasoning model are called *Inquiry Dialogue Frameworks*, and the relevance notion is omitted in their formal specification. In particular, the natural relevance notions associated to Φ^{lp} and Φ^{naf} are noted respectively $\mathcal{N}_h^{lp}$ and $\mathcal{N}_h^{naf}$, and the resulting inquiry frameworks are referred to as $\mathfrak{I}^{lp}$ and $\mathfrak{I}^{naf}$.

Lastly, the authors distinguish two sets of knowledge involved in the dialogue. The *public knowledge* at certain step j, noted $\mathbf{PU}_d^j$, is defined as the union of all the contributions made so far in the dialogue d. The *private knowledge*, noted $\mathbf{PR}_{\mathfrak{F}}$, corresponds to the union of all the private knowledge bases of the agents conforming the dialogue framework $\mathfrak{F}$. Observe the former is dynamic whereas the latter is a static notion.

3 Collaborative Semantics for Abstract Dialogue Frameworks

Given a dialogue framework and a topic, it is interesting to analyze how should the dialogue evolve, and which should be the conclusion. Such analysis is of great relevance for dialogue automatization and evaluation of existent models. Three *collaborative semantics* for dialogue frameworks are defined in [15] describing different ways of solving those questions. All of them pursue the characterization of dialogues achieving the following requirements:

$\mathbf{R}_1$ all the relevant information is exposed,
$\mathbf{R}_2$ the exchange of irrelevant information is avoided and
$\mathbf{R}_3$ the final conclusion follows from all that has been said.

[1] It is assumed that *PLP* contains the following binary connectives: $\wedge$ and $\leftarrow$, and *PLP*$_{naf}$ also includes the unary prefix connective **not** . Examples in *PLP*$_{naf}$ will have exactly one stable expansion, so there will be no confusion regarding their semantics.

3.1 Utopian Collaborative Semantics

The *utopian collaborative (U.C.) semantics* describes an idealistic, in most cases impractical behavior of collaborative agents, relying on the unrealistic assumption that each one is aware of the private knowledge of the others. Its usefulness is thus only theoretical, to be used as a benchmark against which to compare other semantics. It is defined by three conditions: *global completeness* states that no subset of the private knowledge is relevant to the final public knowledge, *global progress* states that each move is part of a (possibly distributed) pure relevant contribution to the public knowledge so far generated, and *correctness*s states that the dialogue outcome is the result of reasoning from the final public knowledge. A relevant contribution is said to be *pure* if there are no irrelevant sentences infiltrated within it (for a detailed definition see [15]). It is clear that these conditions formalize, respectively, the three requirements R_1, R_2 and R_3 mentioned earlier. The formal definition with an illustrative example follows.

Definition 1 (Utopian Collaborative Semantics). *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework. An admissible dialogue $d = \langle t, \langle m_j \rangle, o \rangle$ belongs to the* Utopian Collaborative Semantics *for $\mathfrak{F}$ (noted Utopian($\mathfrak{F}$)) if, and only if:*

Global Completeness: *if m_j is the last move in the sequence, then $\mathbf{PR}_{\mathfrak{F}}$ does not contain any t-relevant contribution to $\mathbf{PU}_d^j$.*

Global Progress: *for each move $m_j = \langle id_j, \mathrm{X}_j \rangle$ in the sequence, there exists $\mathrm{Y} \subseteq \mathbf{PR}_{\mathfrak{F}}$ such that $\mathrm{X}_j \subseteq \mathrm{Y}$ and Y is a pure t-relevant contribution to $\mathbf{PU}_d^{j-1}$.*

Correctness: *if m_j is the last move in the sequence, then $\Phi(\mathbf{PU}_d^j, t) = o$.*

Example 1. *Consider an instance of the $\mathfrak{I}^{lp}$ framework, where the set* Ag *is composed by* $\mathrm{K_A} = \{a \leftarrow b, e\}$, $\mathrm{K_B} = \{b \leftarrow c, b \leftarrow d, f\}$ *and* $\mathrm{K_C} = \{c, g\}$. *The dialogue d_1 shown on the right, over topic 'a', and also all the permutations of its moves with the same topic and outcome, belong to the Utopian Collaborative Semantics for the framework. The chart traces the dialogue, showing the partial results of reasoning from the public knowledge so far generated. The last of these results (underlined) is the final dialogue outcome.*

j	A	B	C	$\Phi(\mathbf{PU}^{\mathrm{j}}_{\mathrm{d}_1}, a)$
1	$a \leftarrow b$			No
2		$b \leftarrow c$		No
3			c	<u>Yes</u>

3.2 Basic Collaborative Semantics

The *basic collaborative (B.C.) semantics* is a practical approach to the utopian behavior, obtained by replacing global completeness and global progress with their local versions: *local completeness* states that no subset of any individual knowledge base is relevant to the final public knowledge, and *local progress* states that each move is purely relevant to the public knowledge so far generated. The formal definition follows.

Definition 2 (Basic Collaborative Semantics). *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework. An admissible dialogue $d = \langle t, \langle m_j \rangle, o \rangle$ belongs to the* Basic Collaborative Semantics *for $\mathfrak{F}$ (noted Basic($\mathfrak{F}$)) if, and only if, the following conditions, as well as* **Correctness** *(Def. 1), hold:*

Local Completeness: *if m_j is the last move in the sequence, then it does not exist an agent* $\mathrm{K}_{id} \in \mathrm{Ag}$ *such that* K *contains a t-relevant contribution to* $\mathbf{PU}_d^j$.

Local Progress: *for each move $m_j = \langle id_j, \mathrm{X}_j \rangle$ in the sequence,* X_j *is a pure t-relevant contribution to* $\mathbf{PU}_d^{j-1}$.

The clear drawback of this approximation is that it misses relevant contributions whenever they are distributed such that none of the parts is relevant by itself, and that means global completeness cannot be guaranteed. This situation is illustrated below.

Example 2. *Consider an instance of the* $\mathfrak{I}^{naf}$ *inquiry framework, where the set* Ag *is composed by* $\mathrm{K_A} = \{a \leftarrow b \wedge \mathbf{not}\ c, b\}$, $\mathrm{K_B} = \{d \leftarrow \mathbf{not}\ e, f\}$, *and* $\mathrm{K_C} = \{c \leftarrow \mathbf{not}\ d, e \leftarrow f\}$. *The dialogue traced on the right, over topic 'a', belongs to the Basic Semantics for the framework instantiated above. Note that* global completeness *is not achieved, since there still exists a distributed relevant contribution when the dialogue ends:* $\{e \leftarrow f, f\}$.

j	A	B	C	$\Phi(\mathbf{PU}_d^j, a)$
1	$a \leftarrow b \wedge \mathbf{not}\ c$ b			Yes
2			$c \leftarrow \mathbf{not}\ d$	No
3		$d \leftarrow \mathbf{not}\ e$		Yes

3.3 Full Collaborative Semantics

The *full collaborative (F.C.) semantics* is an enhancement of the previous (B.C. semantics) in order to cope with the distributed relevance problem. It is obtained by replacing the original relevance notion (say $\mathcal{R}_\mathrm{t}$) with another one (say $\mathcal{P}_\mathrm{t}$) which is called a *potential relevance notion*, such that: every relevant contribution under $\mathcal{R}_\mathrm{t}$ is relevant under $\mathcal{P}_\mathrm{t}$ ($\mathcal{R}_\mathrm{t}$ is said to be *stronger or equal* than $\mathcal{P}_\mathrm{t}$), and also every relevant contribution under $\mathcal{R}_\mathrm{t}$ contains at least one individual sentence which is by itself relevant under $\mathcal{P}_\mathrm{t}$. Clearly, this ensures global completeness under the original notion. The formal definitions follow. Observe this is actually a family of semantics, since each possible potential relevance notion generates a different F.C. semantics.

Definition 3 (Potential Relevance Notion). *Let $\mathcal{R}_t$ and $\mathcal{P}_t$ be relevance notions. We say that $\mathcal{P}_t$ is a* potential relevance notion for $\mathcal{R}_t$ *if, and only if:*

1. $\mathcal{R}_t \subseteq \mathcal{P}_t$ *($\mathcal{R}_t$ is said to be* stronger or equal *than $\mathcal{P}_t$), and*
2. *if* $\mathrm{X}\mathcal{R}_t\mathrm{S}$ *then there exists* $\alpha \in \mathrm{X}$ *such that* $\alpha\mathcal{P}_t\mathrm{S}$.

If $\mathrm{X}\mathcal{P}_t\mathrm{S}$ *and $\mathcal{P}_t$ is a potential relevance notion for $\mathcal{R}_t$, then* X *is said to be a* potential t-relevant contribution to S under $\mathcal{R}_\mathrm{t}$.

Definition 4 (Potential Dialogue Framework). *Let* $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ *and* $\mathfrak{F}^* = \langle \mathcal{P}_t, \Phi, \mathrm{Ag} \rangle$ *be abstract dialogue frameworks. We say that* $\mathfrak{F}^*$ *is a* potential dialogue framework *for* $\mathfrak{F}$ *if, and only if, $\mathcal{P}_t$ is a potential relevance notion for $\mathcal{R}_t$.*

Definition 5 (Full Collaborative Semantics). *Let* $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ *be an abstract dialogue framework. An admissible dialogue* $d = \langle t, \langle m_j \rangle, o \rangle$ *belongs to the* Full Collaborative Semantics *for* $\mathfrak{F}$ *(noted Full($\mathfrak{F}$)) if, and only if,* $d \in Basic(\mathfrak{F}^*)$ *for some framework* $\mathfrak{F}^* = \langle \mathcal{P}_t, \Phi, \mathrm{Ag} \rangle$ *which is a potential dialogue framework for* $\mathfrak{F}$. *We will also use the more specific notation* $d \in Full(\mathfrak{F}, \mathcal{P}_t)$.

The aim of a potential notion is then to detect parts of distributed relevant contributions, or in other words to detect contributions that would be relevant within certain context, being this context determined by other agents' yet unexposed private knowledge. The main drawback of this approach is allowing moves which may not be allowed under the U.C. semantics, since a potential contribution may not actually be part of any distributed contribution (*i.e.,* the required context does not actually exist). That means certain moves may violate global progress under the original relevance notion. As noted in [15], the challenge is then to design *good* potential relevance notions which minimize the amount of such moves. Observe, for instance, that a relevance notion which considers any sentence of the language as relevant, works as a potential for any given relevance notion, but it is clearly not a good one.

Next we show a potential relevance notion for $\mathcal{N}_h^{lp}$ which has been introduced in [15], and which will be reconsidered in Section 4. The notion, called an *abductive relevance notion*, is defined using the concept of abduction in *PLP*. The *abduction set* of a fact h from a set S contains all the minimal sets of facts that could be added to S in order to derive h. This represents possible ways in which the current knowledge could be expanded in order to derive a given fact. The authors then define the abductive relevance notion $\mathcal{A}_h^{lp}$, which considers a contribution to be relevant, wrt. certain topic, if its addition generates a new element in the abduction set of the topic from the public knowledge. This means that either a new potential situation in which the topic would be derived has arisen, or it is actually derived. The formal definitions follow, along with a complete example of a dialogue under the F.C. semantics resulting from this notion.

Definition 6 (Abduction Set). *Let* $\mathrm{S} \subseteq \mathcal{L}^{lp}$ *and* $h \in \mathcal{L}_{Facts}$. *The* abduction set *of* h *from* S *is defined as follows:*

$$AB(\mathrm{S}, h) = \{\mathrm{H} \subseteq \mathcal{L}_{Facts} : (\mathrm{S} \cup \mathrm{H}) \vdash h \text{ and } \nexists \mathrm{H}' \subset \mathrm{H} \text{ s. t. } (\mathrm{S} \cup \mathrm{H}') \vdash h\}$$

Definition 7 (Abductive Relevance). *Let* $\mathrm{S} \subseteq \mathcal{L}^{lp}$ *and* $h \in \mathcal{L}_{Facts}$. *A set* $\mathrm{X} \subseteq \mathcal{L}^{lp}$ *is an* h*-relevant contribution to* S *under* $\mathcal{A}_h^{lp}$ *if, and only if, there exists* $\mathrm{H} \subseteq \mathcal{L}_{Facts}$ *such that: (1)* $\mathrm{H} \in AB(\mathrm{S} \cup \mathrm{X}, h)$ *and (2)* $\mathrm{H} \notin AB(\mathrm{S}, h)$.

Example 3. *Consider the same scenario as in Ex. 1 for the* $\mathfrak{I}^{lp}$ *framework. The dialogue* d_1 *from Ex. 1 belongs to Full*$(\mathfrak{I}^{lp}, \mathcal{A}_h^{lp})$. *Also belongs to this semantics the dialogue* d_2 *traced on the right. The fifth column of the chart shows the evolution of the abduction set of the fact 'a' from the generated public knowledge. An additional step* ***0*** *is added, in order to show the initial state of this abduction set (i.e., when the public knowledge is still empty). Another dialogue under this semantics is the one which results from* d_2 *by interchanging steps* ***2*** *and* ***3****, and also the one which is obtained by merging these two steps together in a single one. Note that all these dialogues achieve global completeness, although global progress is achieved only by dialogue* d_1.

j	A	B	C	$AB(\mathbf{PU}^{\mathrm{j}}_{\mathrm{d}_2}, a)$	$\Phi(\mathbf{PU}^{\mathrm{j}}_{\mathrm{d}_2}, a)$
0				$\{\{a\}\}$	No
1	$a \leftarrow b$			$\{\{a\}\{b\}\}$	No
2		$b \leftarrow c$		$\{\{a\}\{b\}\{c\}\}$	No
3		$b \leftarrow d$		$\{\{a\}\{b\}\{c\}\{d\}\}$	No
4			c	$\{\{\}\}$	Yes

Important results for the dialogues generated under the U.C., B.C. and F.C. semantics have been achieved in [15]. It can be proven that, under certain condition related to

the nature of the relevance notion, the dialogues generated under the three semantics always consist of a finite sequence of steps (*termination*). The necessary condition on the relevance notion, called *novelty*, is that every relevant contribution adds some new information to the public knowledge. Another important result, holding only for those semantics achieving global completeness (*i.e.,* U.C. and F.C. semantics) states that if the relevance notion detects at least all natural contributions (*i.e.,* it is *weaker or equal* than the natural one), then it is possible to determine which the outcome of the dialogue will be, no matter what sequence of steps are actually performed. This last property was called, similarly to other works in the literature, *outcome determinism*. Moreover, this outcome coincides with the result of applying the reasoning model to the private knowledge involved in the dialogue. The authors also state that the natural relevance notion associated to any given reasoning model satisfies the necessary conditions for ensuring the prior results.

4 A Half-Way Collaborative Semantics

So far we have two practical semantics: the basic collaborative one under which distributed relevant contributions are not detected at all, and the full collaborative one under which they are always discovered. In between them, however, there is a gap of unnamed behavior, covering dialogues in which some patterns or cases of distributed relevance are detected but others are not. We propose to name this gap *advanced collaborative (A.C.) semantics*, whose formal definition follows.

Definition 8 (Advanced Relevance Notion). *Let $\mathcal{R}_t$ and $\mathcal{R}_t^+$ be relevance notions. We say that $\mathcal{R}_t^+$ is an* advanced relevance notion for $\mathcal{R}_t$ *if, and only if:*

1. $\mathcal{R}_t \subset \mathcal{R}_t^+$ *($\mathcal{R}_t$ is* strictly stronger *than $\mathcal{R}_t^+$), and*
2. *there exists a contribution* X *under $\mathcal{R}_t$ (*X$\mathcal{R}_t$S *for some* S*) verifying that:*
   (a) *there is no* Y $\subset$ X *such that* Y$\mathcal{R}_t$S *(*X *is minimally relevant under $\mathcal{R}_t$), and*
   (b) *there is a non-empty set* Z $\subset$ X *such that* Z$\mathcal{R}_t^+$S

If X$\mathcal{R}_t^+$S *and $\mathcal{R}_t^+$ is an advanced relevance notion for $\mathcal{R}_t$, we say that* X *is an* advanced t-relevant contribution to S under $\mathcal{R}_t$.

Definition 9 (Advanced Dialogue Framework). *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ and $\mathfrak{F}^+ = \langle \mathcal{R}_t^+, \Phi, \mathrm{Ag} \rangle$ be abstract dialogue frameworks. We say that $\mathfrak{F}^+$ is an* advanced framework *for $\mathfrak{F}$ if, and only if, $\mathcal{R}_t^+$ is an advanced relevance notion for $\mathcal{R}_t$.*

Definition 10 (Advanced Collaborative Semantics). *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework. An admissible dialogue $d = \langle t, \langle m_j \rangle, o \rangle$ belongs to the* Advanced Collaborative Semantics *for $\mathfrak{F}$ (noted Advanced($\mathfrak{F}$)) if, and only if, $d \in$ Basic($\mathfrak{F}^+$) for some framework $\mathfrak{F}^+ = \langle \mathcal{R}_t^+, \Phi, \mathrm{Ag} \rangle$ which is an advanced framework for $\mathfrak{F}$. We will also use the more specific notation $d \in$ Advanced($\mathfrak{F}, \mathcal{R}_t^+$).*

Now we have a formal denotation for the behavior of those agents who make an effort toward achieving global completeness, although they do not always succeed. Considering that potential relevance notions might sometimes be hard to define and/or too

costly in practice to be performed by the agents, such an intermediate semantics could then represent a good compromise between simplicity (B.C. semantics) and sophistication (F.C. semantics). It might be the case, for instance, that almost all patterns of distributed relevance can be easily handled, tough a few of them, which are rarely encountered, require a much more complex treatment. Under this scenario, and assuming the application domain allows for some flexibility regarding requirement R_1 (mentioned in Section 3), then it may be convenient to build an advanced semantics to efficiently cope with the more general cases.

By way of example, we will see that the abductive relevance notion defined earlier in Section 3.3, which works as a potential notion for $\mathcal{N}_h^{lp}$, does not perform so well when applied to $\mathcal{N}_h^{naf}$, however it qualifies as an advanced relevance notion for this last. First we need to generalize the concept of *abduction set* in order to consider the non-monotonicity of PLP_{naf}.

Definition 11 (Abduction Set for PLP_{naf}). *Let* $\mathrm{S} \subseteq \mathcal{L}^{naf}$ *and* $h \in \mathcal{L}_{Facts}$. *The* abduction set *of* h *from* S *is defined as:*

$$AB(\mathrm{S}, h) = \{\mathrm{H} \subseteq \mathcal{L}_{Facts} : \mathrm{H} \textit{ is a minimal natural } h\textit{-relevant contribution to } \mathrm{S}\}$$

In this way, the abduction set points out possible expansions to the current knowledge in order to derive or to stop deriving (whichever applies) the literal under discussion. Now we can introduce an *abductive relevance* for PLP_{naf} analogous to the one presented earlier for *PLP*.

Definition 12 (Abductive Relevance for PLP_{naf}). *Let* $\mathrm{S} \subseteq \mathcal{L}^{naf}$ *and* $h \in \mathcal{L}_{Facts}$. *A set* $\mathrm{X} \subseteq \mathcal{L}^{naf}$ *is an* h*-relevant contribution to* S *under* $\mathcal{A}_h^{naf}$ *if, and only if, there exists* $\mathrm{H} \subseteq \mathcal{L}_{Facts}$ *such that: (1)* $\mathrm{H} \in AB(\mathrm{S} \cup \mathrm{X}, h)$ *and (2)* $\mathrm{H} \notin AB(\mathrm{S}, h)$.

As will be seen in short, most cases of distributed relevance are successfully solved by $\mathcal{A}_h^{naf}$. However, contributions matching a certain problematic pattern remain undetected. The following examples illustrate both situations. Example 4 shows a successful dialogue in which all distributed contributions are revealed, while Example 5 exhibits a difficult case in which global completeness cannot be achieved.

Example 4. *Consider an instance of the* $\mathfrak{I}^{naf}$ *inquiry framework, where the set* Ag *is composed by* $\mathrm{K_A} = \{a \leftarrow b \wedge \mathbf{not}\ c\}$, $\mathrm{K_B} = \{b, d\}$, *and* $\mathrm{K_C} = \{c \leftarrow \mathbf{not}\ d\}$.

j	A	B	C	$AB(\mathbf{PU}_{\mathrm{d}}^{\mathrm{j}}, a)$	$\Phi(\mathbf{PU}_{\mathrm{d}}^{\mathrm{j}}, a)$
0				$\{\{a\}\}$	No
1	$a \leftarrow b \wedge \mathbf{not}\ c$			$\{\{a\}\{b\}\}$	No
2			$c \leftarrow \mathbf{not}\ d$	$\{\{a\}\{b, d\}\}$	No
3		b, d		$\{\{c\}\}$	Yes

The dialogue traced on the right, over topic "a", belongs to the advanced semantics generated by $\mathcal{A}_h^{naf}$. *Observe that the first two are advanced contributions, so they would have not been possible under the basic semantics. In fact, the only possible dialogue under the basic semantics would be the empty one, yielding a wrong outcome. Also observe that all distributed contributions are successfully detected, achieving global completeness and, consequently, outcome determinism. A different dialogue under the same advanced semantics would have had place had fact "b" been exposed in the second move, just after agent* A*'s move. It is easy to see that global completeness would also be achieved in that case.*

j	A	B	C	$AB(\mathbf{PU}^{j}_{d}, a)$	$\Phi(\mathbf{PU}^{j}_{d}, a)$
0				$\{\{a\}\}$	No
1	$a \leftarrow b \wedge \mathbf{not}\ c$			$\{\{a\}\{b\}\}$	No

Example 5. *Consider now that the set* Ag *is composed by* $K_A = \{a \leftarrow b \wedge \mathbf{not}\ c\}$, $K_B = \{a \leftarrow b,\ d\}$, *and* $K_C = \{b \leftarrow c,\ c \leftarrow d\}$. *The dialogue on the right (over topic "a") belongs to the advanced semantics generated by* $\mathcal{A}_h^{naf}$. *Agent* A *makes the first move, an advanced contribution, and after that no other move by any of the agents is possible. Observe that a distributed natural contribution still exists in the private knowledge, although none of the parts is detected by the advanced relevance notion. Global completeness, thus, cannot be achieved, and the dialogue outcome is not the expected one. Note also that the situation would have been different had agent* B, *instead of* A, *made the first move by exposing the rule "*$a \leftarrow b$*". It is easy to see that that event would lead, under the same advanced semantics, to a successful dialogue achieving outcome determinism.*

Finally, some results associated to the A.C. semantics will be formally established. Since this new semantics corresponds to a B.C. semantics applied to an advanced relevance notion, then all the properties achieved by the latter are inherited, provided the chosen advanced notion meets the required conditions. In particular, termination of the dialogues can be ensured as long as the advanced relevance notion satisfies the novelty condition mentioned in Section 3. Formally speaking, a notion $\mathcal{R}_t$ satisfies *novelty* if, and only if, $X\mathcal{R}_tS$ implies $X \not\subseteq S$. The proposition below states this result.

Proposition 1. *Let* $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ *be an abstract dialogue framework, and* $\mathcal{R}_t^+$ *an advanced relevance notion for* $\mathcal{R}_t$. *The dialogues in Advanced*$(\mathfrak{F}, \mathcal{R}_t^+)$ *terminate after a finite sequence of steps, provided that the relevance notion* $\mathcal{R}_t^+$ *satisfies* novelty.

It is clear, and was shown in the previous example, that global completeness is not always satisfied, hence outcome determinism cannot be guaranteed. Although these results do not differ in theory to the ones achieved by the B.C. semantics, the actual difference with respect to this last is that, in practice, there will be situations (at least one) involving the distributed relevance problem, in which the A.C. semantics will behave better. This enhancement is formalized in the following proposition.

Proposition 2. *Let* $\mathcal{R}_t^+$ *be an advanced relevance notion for* $\mathcal{R}_t$, *and* Φ *a reasoning model. There exists a set of agents* Ag, *and a topic t, such that: no dialogue over 't' in Basic*$(\langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle)$ *satisfies global completeness, but there is at least one dialogue in Advanced*$(\langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle, \mathcal{R}_t^+)$ *which does.*

5 Conclusions

We took as starting point an existent multi-agent dialogue framework which provides an abstract environment for studying the collaborative behavior of agents engaged in a dialogue, in terms of any given notion for the relevance of contributions. Three collaborative semantics within this framework (utopian, basic and full) had already been defined in [15] to describe different degrees of collaboration among the agents, with special focus in the handling of the distributed relevance problem.

In this work we have proposed a formalization, under the name of *advanced collaborative semantics*, for the behavior in the transition between the basic and the full degree

of cooperation. To that end, the concept of *advanced relevance notion* has been introduced, which resembles the *potential relevance notion* defined in [15] for building the full collaborative semantics. We have defined a particular instance of this notion using *Propositional Logic Programming with Negation as Failure*, and provided illustrative examples of the generated dialogues. We have also established which properties hold for the dialogues generated under this new semantics, and we have stated a formal result regarding the enhancement of the advanced behavior over the basic one.

Our contribution should be useful for labeling existent approaches which does not fit into any of the previously defined categories, and also for developing new approaches when the full degree of cooperation is too hard to obtain and not really required. An advanced relevance notion might also be used as a starting point toward the construction of a potential one, by means of identification, analysis and resolution of problematic patterns. In that sense, we have left as future work a deeper analysis of the problematic cases related to the abductive relevance notion defined here, to explore possible refinements in order to obtain a potential notion.

References

1. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and Complexity of Some Formal Inter-Agent Dialogues. J. Log. Comput. (2003)
2. Walton, D., Krabbe, E.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany (1995)
3. Kraus, S., Sycara, K., Evenchik, A.: Reaching Agreements through Argumentation: A Logical Model and Implementation. Artificial Intelligence (1998)
4. Amgoud, L., Maudet, N., Parsons, S.: Modelling Dialogues Using Argumentation. In: ICMAS 2000, Boston, USA (2000)
5. Parsons, S., Amgoud, L., Wooldridge, M.: An Analysis of Formal Inter-agent Dialogues. In: AAMAS 2002, Bologna, Italy (2002)
6. Amgoud, L., Parsons, S., Wooldridge, M.: On the Formal Outcomes of Formal Inter-agent Dialogues. In: AAMAS 2003, Melbourne, Australie (2003)
7. Amgoud, L., Prade, H., Belabbes, S.: Towards a Formal Framework for the Search of a Consensus Between Autonomous Agents. In: AAMAS 2005, Utrecht (2005)
8. Amgoud, L., Dimopoulos, Y., Moraitis, P.: A Unified and General Framework for Argumentation Based Negotiation. In: AAMAS 2007, Honolulu, Hawai'i (2007)
9. Black, E., Hunter, A.: A Generative Inquiry Dialogue System. In: AAMAS 2007, Honolulu, Hawai'i (2007)
10. Amgoud, L., de Saint-Cyr, F.D.: Measures for Persuasion Dialogs: A Preliminary Investigation. In: COMMA 2008, Toulouse - France (2008)
11. Parsons, S., McBurney, P., Sklar, E., Wooldridge, M.: On the Relevance of Utterances in Formal Inter-agent Dialogues. In: AAMAS 2007, Honolulu, Hawai'i (2007)
12. Prakken, H.: On Dialogue Systems with Speech Acts, Arguments, and Counterarguments. In: JELIA (2000)
13. Prakken, H.: Relating Protocols for Dynamic Dispute with Logics for Defeasible Argumentation. Synthese (2001)
14. Prakken, H.: Coherence and Flexibility in Dialogue Games for Argumentation. J. Log. Comput. (2005)
15. Marcos, M.J., Falappa, M.A., Simari, G.R.: Semantically Characterizing Collaborative Behavior in an Abstract Dialogue Framework. In: Link, S., Prade, H. (eds.) FoIKS 2010. LNCS, vol. 5956, pp. 173–190. Springer, Heidelberg (2010)

Using Sentence Semantic Similarity Based on WordNet in Recognizing Textual Entailment

Julio J. Castillo[1,2] and Marina E. Cardenas[2]

[1] National University of Cordoba-FaMAF, Cordoba, Argentina
[2] National Technological University-FRC, Cordoba, Argentina
{jotacastillo,ing.marinacardenas}@gmail.com

Abstract. This paper presents a Recognizing Textual Entailment system which uses semantic distances to sentence level over WordNet to assess the impact on predicting Textual Entailment datasets. We extent word-to-word metrics to sentence level in order to best fit in textual entailment domain. Finally, we show experiments over several RTE datasets and draw conclusions about the useful of WordNet semantic measures on this task. As a conclusion, we show that an initial but average-score system can be built using only semantic information from WordNet.

Keywords: Recognizing Textual Entailment, WordNet, Semantic Similarity.

1 Introduction

The Recognizing Textual Entailment (RTE) task [1], [2] is defined as a directional relationship between a pair of text fragments or sentences, called the "text" (*T*) and the"hypothesis" (*H*). Thus, we say that "T entails H", if a human reading *T* would infer that *H* is most likely true.

The two-way RTE task consists of deciding whether:

• *T* entails *H*, in which case the pair will be marked as "Entailment".
• *T* does not entail *H*, in which case the pair will be marked as "No Entailment".

This definition of entailment is based on (and assumes) common human understanding of language as well as common background knowledge, as in the following example.

T1: Internet media company Yahoo Inc. announced Monday it is buying Overture Services Inc. in a $1.63-billion (U.S.) cash-and-stock deal that will bolsterits online search capabilities.
H1: Overture was acquired by Yahoo.

The dataset of <*T,H*> pairs was collected by human annotators. It consists of balanced subsets, which correspond to typical success and failure settings in different applications (Information Extraction, Information Retrieval or Question Answering). Then, the objective of the Recognizing Textual Entailment Challenge is determining whether the meaning of the Hypothesis (*H*) can be inferred from a text (*T*).

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 366–375, 2010.

Recently the RTE4 Challenge has changed to a 3-way task that consists in distinguish among *Entailment*, C*ontradiction* and U*nknown* when there is no information to accept or reject the hypothesis. However the traditional two-way distinction between entailment and non-entailment is still allowed.

In this paper, we used machine learning algorithms and build a set of features WordNet-based, with the aim of drawn some conclusions about the benefit of semantic measures WordNet-based to the RTE task.

Several authors [3, 4, and 5] among others have used WordNet in textual entailment task. However, there is not existing analysis over the most common WordNet measures in order to determinate which metrics are more useful for RTE task. We are interested in measure how WordNet could help in textual entailment task. Thus, in this paper we first propose how to extend the most common word to word WordNet measures to sentence level and then we combine these measures with two existing sentence level measures. Secondly, we assess the impact of these measures in textual entailment task testing over different RTEs training sets.

The remainder of the paper is organized as follows: In section 2, reviews some related work briefly. Section 3 provides a system overview. Section 4 shows how word-to-word semantic measures WordNet-based are extended to sentence level. Section 5 presents the experimental evaluation and discussion of the results. Finally, in section 6 some conclusions are provided.

2 Related Work

There are several measures that have been previously developed with the aim of quantify how two words are semantically related. Some of them are distance oriented measures computed over semantic networks, whereas others are based on probability models learned from large text collections, some are based on information theoretic approaches which exploit the notion of Information Content [6], and also there are hybrid approaches [7]. In this context, we are going to focus only in WordNet semantic measures (ranging in semantic networks and information theory), because regarding the RTE task we observed the common use of WordNet as one of the mainly semantic resource used.

On the other hand, to sentence level, Li et al. [6] proposed recently a procedure to compute semantic similarity between short sentences. In that work the relationship “is a” and “is a part-whole” are used from WordNet in calculating semantic similarity between words based on the shortest path length. Semantic vectors are built and afterwards the cosine distance between them is performed, and did not conducting WSD (Word Sense Disambiguation) for polysemuos words. Li et al. also considers word order similarity. In our approach, we also work to sentence level, conducting WSD. Based on [8], who expressed that syntax plays a subordinate rol for semantics processing of texts, we don't consider syntactic information when computing semantic similarity and WordNet is used in a different way depending on the similarity function chosen in order to calculate a matching algorithm over a bipartite graph (Section 4).

Other methods based on statistical information of words in a huge corpus, such as LSA (latent semantic analysis) have some drawbacks. First, the results are in general,

very sparce sentence vector of hundreds of dimensions [9] and consequently computationally inefficient. Second, a <*T,H*> pair do not have context in a given RTE dataset.

To address these drawbacks, this paper propose some semantic similarity metrics to sentence level computational efficient, based on the resource most widely used in RTE (i.e: WordNet) with the aims of determining the impact of these measures on this task.

3 System Overview

The system is based on a machine learning approach for recognizing textual entailment. Using this approach we tested with different algorithms in order to classify RTE4 and RTE5 test pairs. To deal with RTE4 and RTE5 in a two-way task, we needed to convert this corpus only in two classes: *Yes* (or *Entailment*), and *No* (*Non-entailment*). For this purpose, those pairs present *Contradiction* and *Unknown* were taken as pairs *No*.

The system produces feature vectors for all possible combinations of the development dataset RTE3, RTE4 and RTE5. Weka [10] is used to train classifiers on these feature vectors. We experimented with the following four machine learning algorithms: Support Vector Machine (SVM), AdaBoost (AB), Multilayer Perceptron (MLP), and Decision Trees (DT). The Decision Trees are interesting because we can see what features were selected from the top levels of the trees. SVM, and AdaBoost were selected because they are known for achieving high performances. MLP was used because has achieved high performance in others NLP tasks.

We tested with various parameters (settings) for the machine learning algorithm, such like increasing the confidence factor in DT for more pruning of the trees, different configuration(layers and neurons) for the neural network, and different kernels for SVM. Thus, we make use of classifiers used by other researchers in order to provide a common framework of evaluation.

For both, two and three-way classification task, we used the RTE3, RTE4 and RTE5 datasets from TAC Challenge provided by NIST.

Additionally, we generated the following development datasets named: RTE3-4C and RTE4-4C such as is proposed by Castillo [11] in order to extend the RTEs datasets by using machine translation engines.

4 The Proposed Method

In this paper, we proposed a procedure to compute sentence similarity applied to the <*T,H*> pairs of RTEs datasets computationally efficient $O(n^3)$.

First, we tried to model the semantic similarity of two texts (*T*,*H*) as a function of the semantic similarity of the constituent words of both phrases. In order to reach this objective, we proposed a text to text similarity measure with is based on word to word similarity. We expect that combining word to word similarity metrics would be a good indicator of text to text similarity. The final value of this measure depends on the function which is chosen to compute partial similarities.

Consequently, we are interested in assess whether using WordNet semantic similarity measures can help in textual entailment recognition task or not. For this purpose, we generated a feature vector with six components for both Text and Hypothesis (see subsection 4.1 and 4.2). Every feature is a *finalscore* value (or *MatchingAverage* in the case of longest common substring feature), which is obtained when computing sentence semantic similarity between *T* and *H*.

Thus, we experimented with four WordNet-based measures on the word level, in order to obtain the maximum similarity: Resnik [12], Lin [13], Jiang & Conrath [14] and Pirro & Seco [15], and also considered others two sentence level metrics: one WordNet-based [16] and, the other one, a lexical measure based on Levenshtein distance to compute sentence level similarity [16]. The first four word-to-word measures were taken as base for computing sentence level metric (see Section 4.1), and all of them were selected because of their well know performance in natural language applications and also for their good computational efficiency.

4.1 Extension to Sentence Similarity WordNet-Based

WordNet is used to calculate the semantic similarity between a *T* (Text) and an *H* (Hypothesis). The following procedure is applied:

Step 1. Word sense disambiguation using the Lesk algorithm [17], based on WordNet definitions.

Step 2. A semantic similarity matrix between words in *T* and *H* is defined.

Step 3. A function *Fsim* is applied to *T* and *H*.

Where the Function *Fsim* could be one of the followings:

Step 3.1. The Resnik [12] similarity metric measures the *Information Content* (IC) of the two WordNet concepts by using *LCS*:

$$RES(s,t) = IC(LCS(s,t))$$

where *IC* is defined as:

$$IC(w) = -\log P(w)$$

Information Content (IC) can be considered as a measure that quantifies the amount of information of a concept, and it can be computed based on the probabilities of a concept in a given corpus. The *IC* values are obtained by considering negative the log likelihood, where *P(w)* is the probability of finding the word *w* in a large corpus in English, and *w* is a concept in WordNet. *LCS(s,t)*:is the least common subsume of *s* and *t*.

Step 3.2. The Lin [13] similarity metric, is based on Resnik's measure of similarity, and adds a normalization factor consisting of the information content of the two input concepts *s* and *t*:

$$LIN(s,t) = \frac{2 * IC(LCS(s,t))}{IC(s,t)}$$

Step 3.3. Another metric considered is Jiang & Conrath [14] which is defined as:

$$JICO(s,t) = \frac{1}{IC(s) + IC(t) - 2 * IC(LCS(s,t))}$$

The word similarity measures are normalized in the range [0–1]. The normalization is done by dividing the similarity score provided by the maximum score of that measure.

Step 3.4. The Pirro & Secco [15] similarity metric is also based on Resnik's measure of similarity, but is defined by using information theory and solving a problem with Resnisk's measure when computing the similarity between identical concepts yielding the information content value of their most specific common abstraction that subsumes two different concepts (msca). In the practice, msca gives the most specific common abstraction value for the two given synsets, where the synsets are represented as Lucene documents. Thus, the Pirro-Secco (PISE) similarity metric is the following:

$$PISE(s,t) = \begin{cases} 3 * IC(msca(s,t)) - IC(s) - IC(t) & ,if\ s \neq t \\ 1 & ,if\ s = t \end{cases}$$

Step 4. Finally, the string similarity between two lists of words is reduced to the problem of bipartite graph matching, being performed by using the Hungarian algorithm [18] over this bipartite graph. Then, we find the assignment that maximizes the sum of ratings of each token. Note that each graph node is a token/word of the list.

At the end, the final score is calculated by:

$$finalscore = \frac{\sum_{s \in T, t \in H} opt(Fsim(s,t))}{Max\ (Length(T), Length\ (H))}$$

Where: *opt(F)* is the optimal assignment in the graph.
Length (T) is the number of tokens in *T*, *Length (H)* is the number of tokens in *H*, and $Fsim \in \{RES, LIN, JICO, PISE\}$.

Finally, we noted that the partial influence on each of the individual similarity is going to be reflected on the overall similarity.

4.2 Additional Sentence Metrics

We use two additional metrics such as [16] to compute sentence level values: Levenshtein distance over graphs, and WordNet-based sentence level.

The standard Levenshtein distance quantifies how many changes (character based) are needed to convert one string into another. For example, how many changes are necessary in the hypothesis *H* to obtain the text *T*. For identical strings, the distance is zero. The Levenshtein distance [19] is computed among the characters in the stemmed Text and Hypothesis strings. Text-hypothesis pairs are stemmed with Porter's stemmer[16] and PoS tagged with the tagger in the OpenNLP framework.

Based on Levenshtein distance a sentence lexical distance is defined and the procedure is the following:

- Each string *T* and *H* are divided in a list of tokens/words.
- The similarity between each pair of tokens in *T* and *H* is performed using the Levenshtein distance.
- The string similarity between two lists of words is reduced to the problem of "bipartite graph matching", being performed by using the Hungarian algorithm over this bipartite graph. Then, we find the assignment that maximizes the sum of ratings of each token. Note that each graph node is a token of the list.
  At the end, the final score is calculated by:

$$finalscore = \frac{TotalSim}{Max(Lenght(T), Lenght(H))}$$

Where *TotalSim* is the sum of the similarities with the optimal assignment in the graph. *Length (T)* is the number of tokens in *T*, and *Length (H)* is the number of tokens in *H*.

Other metric that is used to calculate the semantic similarity between a *T* and a *H* is showed in [16]. The following procedure is applied:

Step 1. Word sense disambiguation using the Lesk algorithm, based on WordNet definitions.

Step 2. A semantic similarity matrix between words in *T* and *H* is defined. Words are used only in synonym and hyperonym relationship. The Breadth First Search algorithm is used over these tokens; similarity is calculated using two factors: length of the path and orientation of the path.

Step 3. To obtain the final score, we use matching average.

The semantic similarity between two words (step 2) is computed as:

$$Sim(s,t) = 2 \times \frac{Depth(LCS(s,t))}{Depth(s) + Depth(t)}$$

Where *s* and *t* are source and target words that we are comparing (*s* is in *H* and *t* is in *T*). *Depth(s)* is the shortest distance from the root node to the current node. *LCS(s,t)*:is the least common subsume of *s* and *t*.

The matching average (step 3) between two sentences *X* and *Y* is calculated as follows:

$$MatchingAverage = 2 \times \frac{Match(X,Y)}{Length(X) + Length(Y)}$$

5 Experimental Evaluation

First, we assess the system using cross validation technique with ten folds, in every corpus, testing over the four classifiers for two-way classification tasks. The results are shown in the tables 1 and 2 below. Second, we assess the system to predict RTE4 and RTE5 showing their results in tables 3 and 4. Finally, in table 5 we show the results obtained to predict RTE5 in three-way task. We used the algorithm proposed by Castillo [11] to generate additional training dataset starting from RTE3 and following a double translation process (dtp). Double translation process can be defined as the process of starting with an S (String in English), translating it to a foreign language F(S), for example Spanish, and back again to the English source language $F^{-1}(S)$. We chose Spanish as intermediate language and Microsoft Bing Translator as the only MT system in this process. The augmented corpus is denoted RTE3-4C and it has the following composition in three-way task: 340 pairs "Contradiction", 1520 pairs "Yes", and 1114 pairs "Unknown". On the other hand, the RTE3-4C in two-way task is composed by: 1154 pairs "No", and 1520 pairs "Yes".

We experimented with the following combination of datasets: RTE3, RTE3-4C, RTE4, and RTE5 to deal with two-way and three-way classification task. Four

classifiers are chosen to learn every development set: Support Vector Machine (SVM), Ada Boost (AB), Multilayer Perceptron (MLP) and Decision Tree (ADTree and J48 for two-way and three-way, respectively) using the open source WEKA Data Mining Software [10]. The RTE4 test set and RTE5-gold set were converted to "RTE4 2-way" and "RTE5-2way" taking *contradiction* and *unknown* pairs as no-entailment in order to assess the system in the two-way task. Tables 1 and 2 shows the results obtained with ten folds Cross Validation in two-way task using RTE3 and RTE3-4C, respectively. The features are named as: R(Resnik), L(Lin), J(Jiang), P(Pirro), Lev(lexical distance based on Levenshtein distance), and C(sentence level and WordNet-based [16]). As a notation: All-Lev = represent all features turned-on except Levenshtein.

Table 1. Results obtained with ten folds Cross Validation in two-way task using RTE3

Features	MLP Classifier	SVM Classifier	AB Classifier	DT (ADTree) Classifier
All	**54.75%**	53.37%	**56.25%**	**56.25%**
All-C	53.66%	54.5%	54.33%	50%
RJLP	53.16%	**55.83%**	52.33%	50%
RJL	54%	55.33%	52.83%	50%
RJ	54.5%	53.83%	53.5%	50%
R	49.5%	52.33%	48.83%	50%
J	49.5%	54.5%	54.5%	50%
L	48.5%	51.83%	49.16%	49.5%
P	48.66%	52%	49.67%	50%
W	53.33%	54.33%	50.5%	51.33%
Lev	51.83%	49.66%	54.16%	51.67%
C,Lev	52.67%	54.17%	50.33%	54.83%
All-Lev	53%	54.66%	51.67%	50.66%

Table 2. Results obtained with ten folds Cross Validation in two-way task using RTE3-4C

Features	MLP Classifier	SVM Classifier	AB Classifier	DT (ADTree) Classifier
All	57.41%	55.11%	56.45%	55.98%
All-C	55.07%	51.34%	53.70%	54.50%
RJLP	54.37%	51.17%	51.10%	51.07%
RJL	54.60%	50.94%	51.10%	51.18%
RJ	54.47%	50.47%	51.11%	52.12%
R	51.44%	51%	51.10%	51.61%
J	50.73%	51.10%	51.30%	50.6%
L	51.84%	50.16%	50.60%	50.63%
P	50.87%	51%	51%	50.90%
C	54.94%	54%	**56.55%**	56.38%
Lev	51.81%	50.64%	53.96%	53.83%
C,Lev	55.04%	54.44%	56.89%	**57.09%**
All-Lev	**58.40%**	**55.17%**	56.45%	56.96%

Regarding the tables 1 and 2, we clearly see that adding more features increases the performance of the classification of the employed datasets.

Table 1 shows that the best results are obtained turning-on all features. Indeed, progressive increasing is gained when more features are added. Thus, by using all features, a statistically significant difference is found instead of using individual features isolated.

In Table 2, we found that the best results are obtained with all-features, but turning-off the levenshtein feature, which is the only non-semantic one. However, not statistically significant difference is found with this configuration.

The best performance of our system was achieved with Multilayer Perceptron classifier with RTE3-4C dataset; it was 58.4% of accuracy and turning-off the Levenshtein feature. The average difference between the best and the worst classifier, using all features, was 1.38%, and 2.3% of accuracy.

The following tables 3 and 4 shows the accuracy predicting RTE4, and RTE5 (test set) in two way task. The training set used was RTE3, RTE3-4C, RTE4, and RTE5.

Table 3. Results obtained taking RTE4 as test set in two-way task

Training set	MLP Classifier	SVM Classifier	ADTree Classifier	AB Classifier
RTE3	53.60%	53.84%	52.5%	53%
RTE3-4C	**56.1%**	54.9%	53.4%	53%
RTE5	50%	53.8%	53.8%	53.7%
Baseline	50%	50%	50%	50%

Table 4. Results obtained taking RTE5 as test set in two-way task

Training set	MLP Classifier	SVM Classifier	ADTree Classifier	AB Classifier
RTE3	55.83%	53.83%	50.33%	50.16%
RTE3-4C	**56.83%**	54.17%	51.67%	50.33%
RTE4	54.17%	53.83%	52.5%	52.5%
Baseline	50%	50%	50%	50%

Table 5. Results obtained taking RTE5 as test set in three-way task

Training set	MLP Classifier	SVM Classifier	ADTree Classifier	AB Classifier
RTE3	**52.5%**	50%	50%	50%
RTE3-4C	48.83%	50%	40%	50%
RTE4	50.83%	50%	44%	50%
Baseline	50%	50%	50%	50%

Table 5 shows the accuracy predicting RTE5 in three-way task. The unbalanced proportion of RTE3-4C in three-way task seems to be a problem for the classifiers. We note that the highest accuracy was obtained with Multilayer Perceptron as

learning algorithm and using RTE3-4C as training set.Interestingly, the RTE3-4C training set outperforms the results obtained with RTE3, it shows that by adding more data we gain real value. It seems that contradict to Zanzotto et al, Castillo[20,16] who showed that RTE3 alone could produce higher results that training on RTE3 merged with other RTE datasets. However, the mainly reason of this discrepancy could be how RTE3-4C has been created. It was built based on the same RTE3 and by using machine translation system as a tool to increase the corpus size with the aim of acquiring more semantic variability. Indeed, this result provides additional evidence to support the claim that creating additional dataset with the algorithm proposed by Castillo [11] yields better results.

The results on test set are worse than those obtained on training set, which is most probably due to the overfitting of classifiers and because of the possible difference between these datasets.

The performance in all cases was clearly above those baselines (50% conform a baseline system that always predicts entailment). In the TAC RTE4 Challenge the best score was 74,6% and the worst was 49.7%, with an average score of 58%, which is above that 56% acc obtained with RTE3-4C as training set and MLP as classifier. In the TAC RTE5 Challenge the best and worst score were 73.5% and 50%, with a median score of 61% for two-way task. The scores for three-way task were 68% (highest), 43% (lowest) and 52% (average). In this case, our best results were 56.8% and 52.5% for two and three-way task. It shows an average score for our system in 3-way task, and lower score that the average in 2-way task and they are good results if we have into account that we are only using and evaluating a specific resource (WordNet) and their related measures. In Addition, the 3-way task is quite challenging, as the average score compared with the results achieved in the 2-way task.

Despite the fact that the results are (slightly) under the average accuracy of the other RTE system, we believe that the results obtained are very promising. This is because of the fact that we are not using any other knowledge resource different to WordNet, and also we are not using any other lexical or syntactic information. Surely, to build a more competitive system is needed to incorporate much more lexical, syntactic and semantic information.

Also, we observed that the use of expanded Corpus improves the accuracy of our system, in any configuration desired with 2.5%, and 1% of accuracy points over the best classifier over RTE3, predicting the RTE4 and RTE5.

6 Conclusions

In this work we present how to extend the classic word-to-word measures to a sentence level with a concrete application in the RTE systems. From our experiments we concluded that by using only a set of semantic features based on WordNet it is possible to obtain results closely to the average score of RTE systems. Other important result from our experiments is that all proposed features from WordNet are not enough to build a competitive RTE system, although an average score could be reached. We also conclude that by using a combination of the semantic sentence measure proposed yields the best result in the majority of the cases.

Finally, we show that our results were improved by applying a promising algorithm to expand RTE Corpus.

Future work is oriented to incorporate additional lexical similarities features and semantic resources and to test the improvements they may yield.

References

1. Giampiccolo, D., Dang, H., Magnini, B., Dagan, I., Cabrio, E.: The Fourth PASCAL Recognizing Textual Entailment Challenge. In: Proceedings TAC 2008 (2008)
2. Bentivogli, L., Dagan, I., Dang, H., Giampiccolo, D., Magnini, B.: The Fifth PASCAL Recognizing Textual Entailment Challenge. In: Proceedings TAC 2009 (2009)
3. Herrera, J., Penas, A., Verdejo, F.: Textual Entailment Recognition Based on Dependency Analysis and WordNet. PASCAL. In: First Challenge Workshop (2005)
4. Ofoghi, B., Yearwood, J.: From Lexical Entailment to Recognizing Textual Entailment Using Linguistic Resources. In: Australasian Language Technology Association Workshop (2009)
5. Castillo, J.: A Machine Learning Approach for Recognizing Textual Entailment in Spanish. In: NAACL, Los Angeles, USA (2010)
6. Li, Y., McLean, D., Bandar, Z., O'Shea, J., Crockett, K.: Sentence Similarity based on Semantic Nets and Corpus Statistics. IEEE TKDE 18(8), 1138–1150 (2006)
7. Li, Y., Bandar, A., McLean, D.: An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. IEEE TKDE 15(4), 871–882 (2003)
8. Wiemer-Hastings, P.: Adding Syntactic Information to LSA. In: Proc. 22nd Ann. Conf. Cognitive Science Soc., pp. 989–993 (2000)
9. Landauer, T., Foltz, P., Laham, D.: Introduction to Latent Semantic Analysis. Discourse Processes 25(2-3), 259–284 (1998)
10. Witten, I., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
11. Castillo, J.: Using Machine Translation Systems to Expand a Corpus in Textual Entailment. In: Icetal 2010, Reykjavik, Iceland (2010)
12. Resnik, P.: Information Content to Evaluate Semantic Similarity in a Taxonomy. In: Proc. of IJCAI 1995, pp. 448–453 (1995)
13. Lin, D.: An Information-Theoretic Definition of Similarity. In: Proc. of Conf. on Machine Learning, pp. 296–304 (1998)
14. Jiang, J., Conrath, D.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: Proc. ROCLING X (1997)
15. Pirrò, G., Seco, N.: Design, Implementation and Evaluation of a New Similarity Metric Combining Feature and Intrinsic Information Content. Springer, Heidelberg (2008)
16. Castillo, J.: Recognizing Textual Entailment: Experiments with Machine Learning Algorithms and RTE Corpora. In: Cicling 2010, Iaşi, Romania (2010)
17. Lesk, L.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In: Proceedings of SIGDOC 1986 (1986)
18. Kuhn, H.: The Hungarian Method for the assignment problem. Naval Research Logistic Quarterly 2, 83–97 (1955)
19. Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady 10, 707 (1966)
20. Zanzotto, F., Pennacchiotti, M., Moschitti, A.: Shallow Semantics in Fast Textual Entailment Rule Learners, RTE3, Prague (2007)

Complementing RRL for Dialogue Summarisation*

Norton Trevisan Roman[1] and Ariadne Maria Brito Rizzoni Carvalho[2]

[1] School of Arts, Sciences and Humanities – University of São Paulo, São Paulo, Brazil
norton@usp.br
[2] Institute of Computing – University of Campinas, Campinas, Brazil
ariadne@ic.unicamp.br

Abstract. This paper describes the use of NECA's RRL – Rich Representation Language – in the aid of automatic dialogue summarisation. We start by defining a set of summary acts, and then present a mapping between these acts, taken from human generated summaries, and their corresponding speech acts from the source dialogues. This mapping can then be used by a summary planner to identify portions in the dialogue with higher probability to support some argument it is about to introduce in the summary. Depending on the application, it could also be used to gather, in the source dialogue, specific information to fulfill summary templates. In following this procedure, we hope researchers will be able to build more human-like summaries, by linking corpora summaries to their source dialogues.

1 Introduction

RRL – *Rich Representation Language* [11, 12] – is an XML-based coding language used to represent, amongst other things, the gestures, facial expressions, speech and semantic information exchanged by conversational agents. Originally developed for the NECA platform[1] [4], this language is also designed for representing each speech act in a dialogue (called Dialogue Act in NECA), as well as its semantic content, along the lines of Kamp and Reyle's Discourse Representation Theory (DRT) [6].

The fact that, within NECA, the user can build some Embodied Conversational Agents (ECAs), setting up their personality traits (such as their politeness degree, for instance) and then automatically generating dialogues between these agents, makes it a suitable platform for other dialogue related tasks. One such task is the automatic generation of summaries which, given an RRL, is to produce a summary for the dialogue the RRL represents (in the same representation language). Also, although NECA has been developed to generate dialogues for

* This research was sponsored by CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico – and CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

[1] Net Environment for Embodied Emotional Conversational Agents.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 376–385, 2010.

two distinct domains, in this paper we will only focus on one of them – *eShowRoom* – which illustrates the interaction between a car seller and a prospective buyer.

However useful, the greatest problem with NECA's RRL lies in the fact that, even though each and every utterance is assigned a dialogue act (*e.g. inform*, *greeting* etc), some of them have no explicitly defined semantic content. As a matter of fact, in the *eShowRoom* scenario there seems to be only two instances where such information is found[2] (see [8]), to wit, (a) when some of the car characteristics are informed or asked for; or (b) when some specific topic is brought up for discussion, as in "I'd like to know more about this car". This limitation turns out to be rather damaging for any automatic system that must rely on a deeper semantic analysis of the dialogue to build its outputs.

In this paper, we describe our attempt to address the above limitation, by mapping the semantic content of human generated summaries (all abstracts), taken from previous research [14–16], to their counterparts in the RRL of the source dialogues. To do so, we first define a set of summary acts – those acts human summarisers seem to perform when building dialogue summaries – and then annotate summaries accordingly. Next, we relate each of these acts to their corresponding dialogue acts in the source dialogue, creating a mapping. Differently from current literature (*e.g.* [2, 3]), however, which deals almost exclusively with lexical and syntactical information, our approach works at the semantic level, thereby making it possible to produce summaries in a language different from that of the summarised dialogue.

With this mapping at hand, automatic summarisers, during the document planning stage, could follow the links between the information they intend to add to the summary, and those spots in the source dialogue which are more likely to support that information. In doing so, they would be able, for example, to backtrack and build a new plan for the document, should the dialogue fail to back it up, thereby overcoming the difficulty the lack of a deeper semantic representation for the dialogue poses to the summarisation process[3]. Depending on the summarisation method, the same mapping could also be used to mine, in the source dialogue, specific information needed to fulfill some summary template, for instance (*cf.* [5, 7, 13]).

The rest of this paper is organised as follows. Section 2 describes a set of summary acts we have defined to represent part of the summary semantics. It also describes the corpus annotated with this information, in order to build the mapping between the semantics of human-produced summaries and their source dialogues. Next, in Section 3, we determine those summary acts that were more used by human summarisers, providing an overview of their importance. Section 4, in turn, presents the resulting mapping, identifying which summary acts were linked to just a few dialogue acts, and which might be supported by a good deal of them. Finally, Section 5 presents a conclusion to this work.

[2] The reason why this is so were left unspecified in NECA's documentation.

[3] These were the actual steps followed when building our own automatic dialogue summariser, described in [16].

2 Corpus Annotation: Extending the RRL

Our first step towards building a mapping that links the content of summaries to their source in the summarised dialogues was to start determining its semantic content, by defining a set of summary acts. Based on Searle's definition of speech acts [17], a summary act represents the illocutionary act (*cf.* [1]) executed by the summariser when writing some clause in the summary, *i.e.* the act underlying that clause. In this research, clauses, *i.e.* a unit consisting, as a minimum, of a verb and its complements [9], were taken as our basic unit due to the relative objectivity of this concept.

To get to this set of summary acts, we analysed 240 summaries (1,773 clauses) from the corpus described in [14, 15], leading to a total of nine summary acts, shown on Table 1. Each clause from the corpus was then annotated with a summary act and a predicate-arguments pair (in first-order logic), representing the clause's semantic content. The construction of each pair builds on the theory presented by Parsons [10], according to which the information presented in a simple sentence – called eventuality – can be classified either as (i) a state, describing some characteristic of an object or person; something that, at a specific moment, is simply true or false (*E.g.* "He is a good seller"); (ii) an event, presenting some fact, action or even state change (*E.g.* "He grew nervous" and "The buyer walked into the shop"; or (iii) a process, *i.e.*, an event with no apparent finishing point, composed of various sub-events identical to the main one, as in "He kept on babbling about it".

Based on the above distinctions, each clause in the corpus was assigned a set of predicates, joined by the symbol '&', so that each predicate represents a single eventuality in that clause. To do so, predicates are named after the verb (either single or complex), in the case of events and processes. Along with this information, they are also assigned a list of complementary attributes, built from the remaining elements in the clause. As for states, we kept NECA's representation, naming the predicate as "attribute" and adding to its complementary list the attribute of the object or participant it deals with. This list, whenever possible, should contain information about:

- Who executed the action (for events or processes), or bears a determined attribute (for states), as in "*The vendor* politely showed the car to the customer", or "It was a very good *car*". This information is compulsory.
- To whom the action was directed, as in "The vendor politely showed the car to *the client*". This information does not apply to states, being optional for the rest (*i.e.*, it is included only if found in the clause).
- What object was involved, as in "The vendor politely showed *the car* to the client". This information does not apply to states and, depending on the situation, can be optional for the rest.
- What modifier was used, as in "The vendor *politely* showed the car to the customer". This information is optional.

Under this classification, clauses like "she (buyer) asked me (vendor) about the luggage compartment" become *ask(buyer, vendor, luggage_compartment)*,

Table 1. Summary acts defined for this research

Summary Act	*Description*
Advice	When the summariser, by writing the clause, advises the reader to do something. *E.g.* "If you ever need a new car, *you should see him*".
Evaluation	When the clause, either directly or indirectly, evaluates some product or dialogue participant; or else when it describes an evaluation by some participant. *E.g.* "Ritchie was ambiguous", "Tina did not like the car", or "It seemed to be a good car".
Auxiliary	Either used with interjections, or with clauses whose semantic content acts solely as a complement to some other clause's semantics, as in "*It was precisely* what I was looking for" (Auxiliary + Evaluation).
Closure	When the clause describes the way the dialogue finished. *E.g.* "It was a smooth sell", "Still I bought the car".
DescrSituation	When the clause summarises the entire dialogue. *E.g.* "This is a dialogue between a vendor and a buyer".
Inform	When the clause only informs about some object's characteristic. *E.g.* "The car's got ABS".
InformAction	When the clause reports on some action executed by some participant. *E.g.* "I informed the client (...)" (incomplete) and "Ritchie described some positive points of that car to Tina".
Opening	The clause describes the way the dialogue began, or the situation that happened at its beginning. *E.g.* "The buyer asked about a car" and "Tina walked into the shop".
Opinion	When the clause explicitly presents the summariser's personal opinion. *E.g.* "Although I find it strange only 100 mph for a sport car". Through this act, the summariser explicitly shows him/herself to the reader.

whereas "An argumentative client came up to the shop" becomes *come_up(buyer, shop)& argumentative(buyer)*. Even though this codification can represent with reasonable precision the meaning of a clause, it fails in capturing its emotional load (a necessary feature to deal with the data from [15]). This limitation was nevertheless addressed by adding an extra predicate to the clause's meaning, responsible for the identification of its polarity (positive or negative), along with the loaded element (either attribute or predicate). As such, clauses like "At least she (buyer) was patient" can be codified as *patient(buyer)&positive(patient)*.

3 Corpus Analysis

By analysing the density of each summary act within summaries, *i.e.*, the mean number of clauses per summary labelled with that act (Figure 1), along with the overall proportion of clauses labelled with them (*i.e.* their coverage, shown on Figure 2), we can see that some summary acts were considered more relevant than others. One such case relates to reporting on someone's behaviour, *i.e.* the action s/he performed (represented by *informAction*). This act's density was

almost twice as high as the second-ranked one, being found in almost 34% of all 1,773 clauses in our corpus, which is a good indication of the importance of such information to the summary [16]. At the other end of the scale, the least used *Acts* where those related either to overviews of the entire situation (*DescrSituation*), or to the personal opinions summarisers add to the summary, both explicitly (*opinion*) and indirectly, through some advice given to the reader (*advice*).

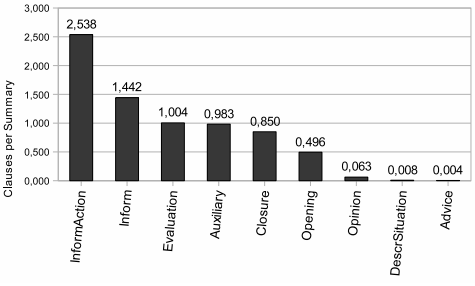


Fig. 1. Summary Acts density: Number of clauses per summary

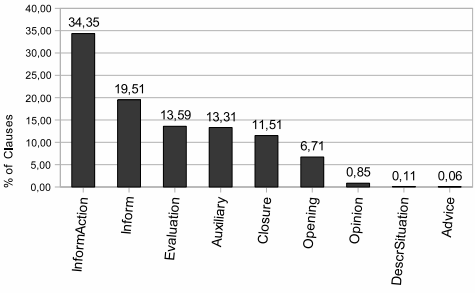


Fig. 2. Clause coverage for the Summary Acts

Interestingly, the summary act *inform*, which conveys purely technical information about the negotiated object, comes in second place in both analyses, indicating that, although not the most widely used, such information still plays, as we might expect, a prominent role in the summaries. If, however, we move to the analysis of each act's coverage amongst summaries, *i.e.* the proportion of summaries with some clause labelled with it, distribution changes considerably, specially regarding to the purely technical information (Figure 3). When comparing both clause and summary coverage, we see this information moving from second down to sixth place, being found in 32.5% of all summaries. This result indicates that, although some people might have considered it paramount, as reflected by the amount of clauses labelled so, the majority of human summarisers decided to let it out of their summaries.

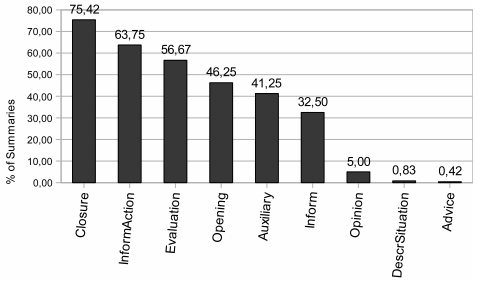


Fig. 3. Summary coverage for the Summary Acts

On the other hand, some *Acts*, such as *closure* for instance, moved from the fifth position (11.5%), in the clause coverage rank, to first amongst dialogues (75.4%). This is somewhat expected, since people are supposed to mention the way the dialogue was closed by its participants, while not bothering to talk too much about it. Similar results can be observed for *opening*, which moved from sixth (6.7%) to fourth position (46.3%). Finally, *Acts* responsible for subjective or behaviour-related information, such as some *evaluation* and *informAction*, were not so much affected by moving from clause to summary distribution (although they could still be found in over 50% of both sets).

4 Mapping Summaries to Dialogues

Once determined the semantics of all clauses in the summaries, our next step was to link those clauses to their counterparts in the source dialogue. Ideally, the automatic summariser should look directly in the source dialogue for the desired propositional content. However, as illustrated on Table 2 (taken from [8]), since over 50% of NECA's utterances present no other semantics apart from their dialogue acts, there usually is no straightforward way of automatically determining the propositional content of all utterances in the source dialogue. Hence, the only alternative we are left with is to handcraft a mapping between each summary act (from the summary clauses) and all dialogue acts (in the source dialogue) that might support that clause. Such a mapping could then be used by the automatic summariser to choose, amongst all competing propositional contents taken from the human produced summaries, the most probable for the dialogue at hand, given the summary act elected by the planner to take part in the final summary.

As expected, the resulting mapping has, as one of its characteristics, the existence of many-to-many relations, *i.e.*, situations in which each summary act can relate to many different dialogue acts (and vice-versa), as shown on Figure 4[4]. Another interesting point about this mapping is the considerable

[4] *Auxiliary* was left out of this figure because it is not directly related to the summary, but the summariser instead.

Table 2. Dialogue acts in NECA

Dialogue Act	*Context*	*Contains Semantics*
agree	The vendor agrees with some negative comment from the buyer	sometimes
initiateClosingNegative	The buyer rejects the car, closing the conversation	no
completeClosingNegative	Vendor's answer to initiateClosingNegative	no
initiateClosingPositive	The buyer takes the car, closing the dialogue	no
completeClosingPositive	Vendor's answer to initiateClosingPositive	no
confirm	The vendor confirms some yes/no question	no
disagree	The vendor disagrees on the buyer's negative remark	no
disconfirm	The vendor denies some yes/no question	no
feedback	Client's reaction, in the absence of a positiveResponse or negativeResponse	no
greeting	The vendor's greetings to the customer at the beginning of the dialogue	no
inform	The vendor informs the buyer about some attribute	yes
negativeEvaluation	The vendor informs the buyer about some negative feature	yes
negativeResponse	The buyer makes a negative remark on some attribute	yes
openingComplaint	The buyer starts the dialogue with a complaint	no
openingComplaintResponse	Vendor's answer to an openingComplaint	no
openingQuestion	The vendor poses a question to start the conversation	no
openingResponse	Buyer's answer to openingQuestion	yes
positiveEvaluation	The vendor informs the client about some positive attribute	yes
positiveResponse	The buyer makes a positive remark on some attribute	sometimes
refuseAnswer	The vendor refuses to answer some question	no
refuseAnswerResponse	Buyer's answer to refuseAnswer	no
requestIf	The buyer poses a yes/no question	yes

Table 2. *(continued)*

Dialogue Act	*Context*	*Contains Semantics*
requestInfo	The buyer's first question	yes
requestValue	The buyer asks for some specific attribute	yes

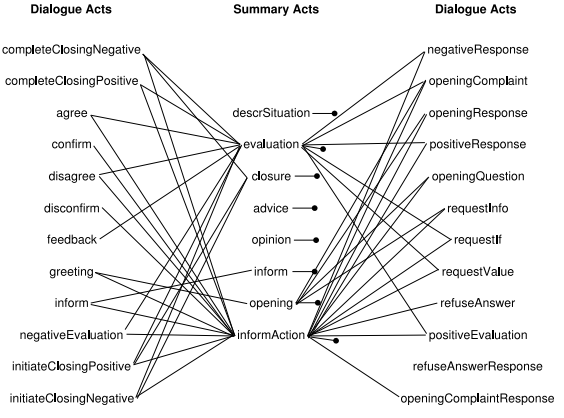


Fig. 4. Mapping between summary acts and dialogue acts

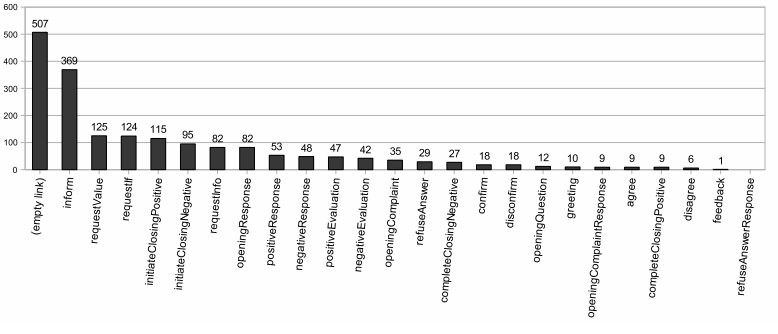


Fig. 5. Distribution of links amongst targeted Dialogue Acts

amount of summary acts with no direct link to some dialogue act (around 27% of all links), represented by the symbol —• in the figure. In this case, the absence of such a link can be interpreted not as the inclusion of information from outside the dialogue, but instead, as the inclusion of information that, although present in the dialogue, has no explicitly defined semantic content within its RRL representation, *i.e.*, it neither has a predicate-arguments pair nor

a dialogue act assigned to. At the other end, we have refuseAnswerResponse with no link whatsoever, meaning that no human summary reported on its content. Figure 5 summarises these results.

5 Conclusion

In this paper we described the measures taken to use RRL in the process of dialogue summarisation. Such measures included (i) the identification and definition, from experimental data, of a set of summary acts used to annotate these data with their semantic meaning; and (ii) the mapping between these summary acts and the *Dialogue Acts* produced by the NECA platform. Differently from current literature (*e.g.* [2, 3]), however, our approach works at the semantic level only, instead of relying on syntactical features.

With this mapping at hand, automatic summarisers could identify, at the document planning stage, those portions in the dialogue with higher probability to support some argument, or even to gather, in the source dialogue, specific information to fulfill some predefined templates (as it was already done in [16], with very encouraging results). As an additional benefit, the method can also be used to draw some conclusions, from experimental data, about the importance of certain summary acts within specific domains, thereby influencing the way automatic summarisers choose (or learn to choose) their output summary content.

Avenues for future work include determining to what extent is our inventory of summary acts specific to the car sales domain. It would be also interesting testing this approach on different domains, such as NECA's *Socialite* scenario, for example, extending the already defined set of *Summary Acts* if necessary. Finally, since the summary acts were determined exclusively by one of the researchers, it is necessary to test them with other people, comparing the obtained results.

References

1. Austin, J.L.: How to do Things with Words, 2nd edn. Oxford University Press, Oxford (1975)
2. Banko, M., Mittal, V., Kantrowitz, M., Goldstein, J.: Generating extraction-based summaries from hand-written summaries by aligning text spans (pacling 1999). In: Proceedings of the Pacific Association for Computational Linguistics, Waterloo, Canada (1999)
3. Daum-III, H., Marcu, D.: Induction of word and phrase alignments for automatic document summarization. Computational Linguistics 31(4), 505–530 (2005)
4. van Deemter, K., Krenn, B., Piwek, P., Klesen, M., Schröder, M., Baumann, S.: Fully generated scripted dialogue for embodied agents. Artificial Intelligence 172(10) (June 2008)
5. Kameyama, M., Kawai, G., Arima, I.: A real-time system for summarizing human-human spontaneous spoken dialogues. In: Proceedings of the 4th International Conference on Spoken Language (ICSLP 1996), Philadelphia, USA, vol. 2, pp. 681–684 (1996)

6. Kamp, H., Reyle, U.: From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Springer, Heidelberg (1993)
7. Kipp, M., Alexandersson, J., Reithinger, N.: Understanding spontaneous negotiation dialogue. In: Proceedings of the IJCAI 1999 Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Stockholm, Sweden, pp. 57–64 (1999)
8. Klesen, M.: Dialogue act specification for the neca domains. Manuscript, DFKI (June 2002)
9. Miller, J.: An Introduction to English Syntax. Edinburgh University Press Ltd., Edinburgh (2002) ISBN 0 7486 1254 8
10. Parsons, T.: Events in the Semantics of English: A Study in Subatomic Semantics. MIT Press, Cambridge (January 1994) ISBN 0-262-66093-8
11. Piwek, P.: NECA deliverable D3a: Specification of scene descriptions for the neca domains. Tech. rep., ITRI – University of Brighton, Brighton, UK (2002); nECA IST-2000-28580 Deliverable D3a
12. Piwek, P., Krenn, B., Schröder, M., Grice, M., Baumann, S., Pirker, H.: RRL: A Rich Representation Language for the description of agent behaviour in NECA. In: Proceedings of the AAMAS Workshop on Embodied Conversational Agents - Let's Specify and Evaluate them!, Bologna, Italy (2002)
13. Reithinger, N., Kipp, M., Engel, R., Alexandersson, J.: Summarizing multilingual spoken negotiation dialogues. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL 2000), Hong Kong, China, pp. 310–317 (2000)
14. Roman, N.T., Piwek, P., Carvalho, A.M.B.R.: Politeness and summarization: an exploratory study. In: Qu, Y., Shanahan, J., Wiebe, J. (eds.) Papers from the 2004 AAAI Spring Symposium: Exploring Attitude and Affect in Text: Theories and Applications (AAAI-EAAT 2004), March 22-24, pp. 112–117. The AAAI Press, Palo Alto (2004)
15. Roman, N.T., Piwek, P., Carvalho, A.M.B.R.: Politeness and Bias in Dialogue Summarization: Two Exploratory Studies. In: Computing Attitude and Affect in Text: Theory and Applications. The Information Retrieval Series, January 9, vol. 20, pp. 171–185. Springer, Netherlands (2006)
16. Roman, N.T., Piwek, P., Carvalho, A.M.B.R.: Emotion and behaviour in automatic dialogue summarisation. In: Proceedings of the Sixth Workshop in Information and Human Language Technology (TIL 2008), Vila Velha, Brazil, October 27-28, vol. 2, pp. 304–309 (2008)
17. Searle, J.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge (1969)

A Multi-dimensional Annotation Scheme for Behaviour in Dialogues*

Norton Trevisan Roman[1] and Ariadne Maria Brito Rizzoni Carvalho[2]

[1] School of Arts, Sciences and Humanities – University of São Paulo, São Paulo, Brazil
norton@usp.br
[2] Institute of Computing – University of Campinas, Campinas, Brazil
ariadne@ic.unicamp.br

Abstract. In this paper we introduce a multi-dimensional annotation scheme for emotional and behavioural assessment in dialogue summaries. To test the soundness both of the annotation scheme and corresponding guidelines, reliability studies with nine independent annotators were carried out. As an illustration of the utility of our scheme, we have applied it to an already published study and verified whether the same conclusions hold. We hope that, in using our scheme, researchers will be able to save a lot of time and effort that, otherwise, would be spent in planning, developing and testing a scheme of their own.

1 Introduction

Despite the growing interest in emotion and sentiment analysis, recent work on this subject seems to focus mainly on evaluations (*e.g.* [9,3]) and the identification of the semantic orientation of words (*e.g.* [24,10]). With just a few exceptions (*e.g.* [19]), current research seems not to try to identify and classify behaviour and, more specifically, those actions people take that may raise some emotion either on the reader side, as in a text, or on some other conversational party, as in a dialogue.

On this account, empirical evidence for the importance of reporting behaviour and emotion in dialogue summaries has been nevertheless brought up to the community [18]. According to these results, whenever a dialogue presents a very impolite behaviour, as exhibited by any of its parties, human summarisers will tend to report that behaviour in the dialogue's summary, biasing it according to the assumed point of view, *i.e.*, the way some party was reported, which actual party was reported, and even if behaviour should be reported at all, depended to a great extent on the point of view taken by the summariser. Given these results, it becomes clear that automatic dialogue summarisers should deal with this kind of information, were they to build more human-like summaries.

* This research was sponsored by CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico – and CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 386–395, 2010.

Currently, the most feasible alternative available to this end is to apply some corpus linguistics techniques on annotated data sources. To do so, however, it becomes necessary to develop annotation schemes capable of identifying the desired information in a reliable way. As such, in this paper we present a multi-dimensional annotation scheme developed to identify and classify information about emotional behaviour and bias in dialogue summaries. Tests with the scheme were run in a corpus composed of 240 dialogue summaries, produced by 30 independent summarisers [18]. In this corpus, summaries are separated both according to their viewpoint (customer, vendor or observer) and size constraint (unrestricted or limited to 10% of the number of words in the summarised dialogue). Source dialogues were automatic generated by NECA[1] – a platform for automatic generation of dialogues between conversational agents [7] – taking place in a car sales setup where some vendor tries to sell a car to a customer.

In order to demonstrate the usefullness of our annotation scheme, we have applied it to the same corpus used to test it, in an attempt to reproduce the experimental conclusions from [18]. In doing so, we hope to show other researchers the possibilities of our scheme, helping them to carry out their researches about emotion/behaviour reporting and bias without having to spend so much time and effort in developing and testing an annotation scheme of their own. The rest of this paper is organised as follows. Section 2 presents our annotation scheme, describing all dimensions that comprise it. The procedure followed by annotators to apply the scheme in the test corpus, along with its evaluation, is described in Section 3. Our conclusions to this work and venues for future research are presented in Section 5.

2 The Multi-dimensional Annotation Scheme

Following McKeown [14], in this research we take the clause as our basic unit, *i.e.* a unit consisting, as a minimum, of a verb and its complements [15]. As such, by following our scheme, every clause in a summary gets classified according to five distinct dimensions (Figure 1), which try to determine what (or who) was reported (or evaluated) in the clause, by whom and how. These dimensions are:

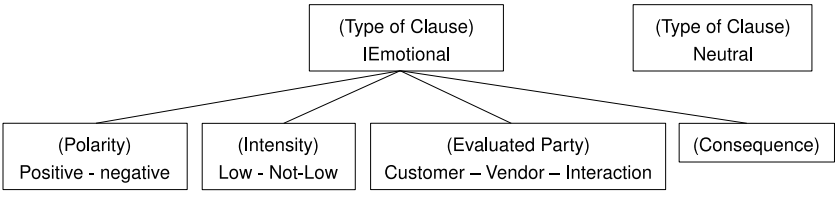


Fig. 1. Dimensions hierarchy in the annotation scheme

[1] Net Environment for Embodied Emotional Conversational Agents.

- *Type of Clause*: following Batliner *et al.* [4] and Fischer [8], this dimension distinguishes between *IEmotional* and *Neutral* clauses. In our research, however, we do not consider a clause as *IEmotional* whenever it presents some emotional feature but, instead, only when it assesses, either positively or negatively, the dialogue participants' politeness degree (more specifically, what Mills [16] called social politeness, or political behaviour), behaviour, humour, feelings and emotions towards some interactional feature (akin to what Keenan *et al.* [11] called the interactional content of the clause), or even when it evaluates the interaction's outcome as a whole.
- *Polarity*: this dimension results both from the theory proposed by Ortony, Clore and Collins [17] and the evaluation proposed by Martin [13], taking only two possible values – *Positive* or *Negative* – that must be applied only to *IEmotional* clauses. Under this dimension, a clause is classified as *Positive* when it describes pleasant actions or feelings, or even when it assesses them so, like in "She was patient" and "The service was fine". On the other hand, it is considered *Negative* when it describes actions or feelings that do not satisfy (and so must be avoided), or when it judges them so, like in "He is rude" or "Tossed all her insatisfaction on me".
- *Intensity*: designed to capture the way perceived emotions or behaviours were described, *i.e.*, towards the lower or higher end in an intensity scale [13], this dimension complements Polarity, in that it tries to determine whether some report was put in a considerably mild way, sounding almost like an euphemism (*Low* intensity) or, rather the opposite, if that demonstration was considered normal, or even above normal, indicating that what was described was taken as something important by the summariser (*Non-low* intensity).
- *Evaluated Party*: this dimension seeks to capture the dialogue participant whose behaviour or emotion was reported either explicitly, like in "the vendor treated me very well", or implicitly, like in "the service was very good", in which case a service necessarily implies a server. Since this dimension was designed primarily for sales dialogues, it takes only one of three possible values: (a) *Vendor*, meaning that the evaluation was made on the vendor's emotions and behaviour; (b) *Client*, when it accounts for the client's; and (c) *Interaction*, which must be used whenever the clause does not assess either of the dialogue participants but, instead, their interaction with each other (directly or indirectly), without particularly focusing in any of them, as in "That was a great sell"[2].
- *Consequence*: must be applied to *IEmotional* clauses describing situations or feeling that were caused by something described in another clause, as in "I was so badly served *that I lost my nerve*", where the later ("I lost my nerve") describes a consequence of the former ("I was so badly served"). More than determining a cause, however, this dimension aims to capture

[2] Clauses like this, however, cannot be taken as *IEmotional* if the reason for the sell being great lies solely on non interactional features, such as a good profit, for example.

the possible ways in which blame is transferred, by having one party justify his/her negative actions on the basis of the other party's behaviour.

Finally, clauses with multiple evaluations, such as "I gently served the rude client", for example, can take multiple classifications too. In this case, the clause may be classified as *IEmotional*, with a positive evaluation about the *Vendor* and a negative evaluation about the *Customer*, both with some *Intensity*.

3 Corpus Annotation and Evaluation

In order to test the soundness both of our annotation scheme and corresponding guidelines [2], we had nine independent annotators apply the scheme to a set of 240 human crafted summaries, comprising 1,773 clauses [18]. As an additional measure to increase reliability [12,2], annotators had to independently go through a training period of about one and a half hour. During this period, they were given a description of the annotation scheme, along with a set of guidelines to help them understand what categories meant (*cf.* [5]). They were then asked to annotate a set of 18 test summaries, with 128 clauses in total, artificially built for this task. Test summaries were written so as to provide a number of specially designed examples, making it easier for annotators to get the grips with the task.

In this research, we assessed our annotation scheme by measuring its reproducibility, *i.e.*, the extent to which annotators will produce the same classifications, working under varying conditions, at different times and locations [12]. To do so, we relied on Krippendorff's alpha as our coefficient of agreement. Ranging from -1 to 1, with 1 meaning total agreement and 0 implying that analysed data cannot be told apart from random events, α can be defined as

$$\alpha = 1 - \frac{D_o}{D_e}$$

where D_o stands for the number of observed disagreements and D_e represents the number of expected disagreements by chance.

When it comes to subjective assessments, however, as is the case with emotion, behaviour and politeness, which are not so clearly defined, perfect agreement becomes much harder to be achieved [20,23,1]. Not to mention that annotating data depends to a great extent both on the personality and humour of its executors [1], which may vary considerably from time to time. In order to account for the effects of this subjectivity in data annotation, we decided to keep the number of categories minimum, by giving annotators fewer options[3].

Also, and as an attempt to reduce the cognitive load on annotators, we have unified dimensions *Type of Clause*, *Polarity* and *Evaluated Party* into a single dimension, thereby providing annotators with a broader picture of the annotation process, as opposed to having them separately focusing on three different dimensions and their categories. As such, instead of choosing, for example,

[3] This measure was inspired on an experiment carried out by Craggs e Wood [6], in which it was observed that adding a single category to a subject dimension dropped agreement from $\alpha = .52$ to $\alpha = 0.37$.

"IEmotional" for *Type of Clause*, "Positive" for *Polarity* and "Vendor" for *Evaluated Party*, annotators were shown "Positive Report about the Vendor" as an alternative. On this matter, it is worth noticing that this unification was made at the user interface level only, *i.e.*, since from "Positive Report about the Vendor" we can independently acquire the value for the three dimensions underneath, the overall model remains untouched.

Results for our annotation scheme are shown on Table 1. Assuming $\alpha > 0.8$ as good reliability, with $0.67 < \alpha < 0.8$ allowing tentative conclusions [12], only *Polarity* turns out as a reliable dimension, with *Type of Clause* and *Evaluated Party* coming next, at the tentative side. Although such values may seem rather disappointing, they are actually good, when compared to current results on emotion classification, which sometimes deliver around $\alpha = 0.6$ as their highest value (*cf.* [6]). Also, the existence of a "Neutral" category which, however necessary, has been described as a common source of confusion [1], just makes it harder to expect a high agreement on dimensions like *Type of Clause*, for example.

Table 1. Alpha values for the annotation

Dimension	α	*Dimension*	α
Polarity	0.843	Intensity	0.212
Evaluated Party	0.783	Consequence	0.085
Type of clause	0.674		

At the other end of the scale, *Intensity* and *Consequence* were found absolutely unreliable. That *Intensity* would score low was somewhat expected, given its variable definition amongst people [17]. As for *Consequence*, however, results showed people do not agree on the causes for the clause they were analysing. Actually, the figures do not get much better if we rule out the specific clause taken for a cause, *i.e.*, if we only keep information about whether some clause was considered as a consequence of another or not. In that case, agreement only becomes slightly better, at $\alpha = 0,175$, indicating that the very notion of cause/consequence, when it comes to emotional assessment, may be rather obscure.

4 Using the Scheme: A Practical Example

To illustrate a possible use of our annotation scheme, we applied it to the database described in [18], so as to identify reports on emotion and behaviour in dialogue summaries, also determining the existence of bias in these reports. Our utmost goal, with this example, is to demonstrate that, through our scheme, the same conclusions can be drawn, *i.e.*, we can still state that (i) if a dialogue contains very impolite behaviour, this behaviour tends to be reported in the dialogue summary; (ii) these reports are biased towards the point of view taken by the summariser; and (iii) severely restricting the summary length has no influence on the previous hypotheses. To do so, our first step was to assign to each

clause in the dataset the label given by the majority of annotators (*cf.* [21]). By way of caution, however, we found it more appropriate to consider majority only those labels chosen by over 50% of all annotators, having ties solved by one of the researchers, who should opt for one of the competing labels[4] (*cf.* [20,1]).

Notwithstanding, before proceeding with the data analysis we must define the categories used by [18] in terms of those of our scheme. In [18], human generated summaries were classified according to one out of three categories: *behav (excl)*, meaning that the summary solely comprised reports on some party's behaviour, thereby ignoring all technical information; *behav*, to be used with summaries delivering both reports on behaviour and technical information; and *non-behav*, representing those summaries dealing exclusively with the technical information exchanged by the dialogue participants. In that research, summaries were also grouped according to the point of view under which they were built and separated according to the maximum amount of words that summarisers were allowed to use[5].

Given this classification, we can connect our research to that of [18] by taking *behav (excl)* summaries to be those summaries composed uniquely of *IEmotional* clauses (as captured by dimension *Type of Clause*), whereas *non-behav* would deliver only *Neutral* clauses and *behav* would be a mix of both. Figure 2 illustrates the results from applying our annotation scheme to the dataset with no constraint on summary size, with summaries grouped according to the source dialogue (from 1 to 4) and point of view under which they were built (customer, vendor or observer). In this figure, we notice that the amount of *behav* summaries is considerably higher for the dialogues portraying impolite behaviour, *i.e.* dialogues 2 and 3 (83% and 97%, respectively), than for those more neutral (dialogues 1 – 37% – and 4 – 50%). This is a statistically significant difference ($\chi^2(1, N = 120) = 29.40$, at the significance level of $p = 0.001$).

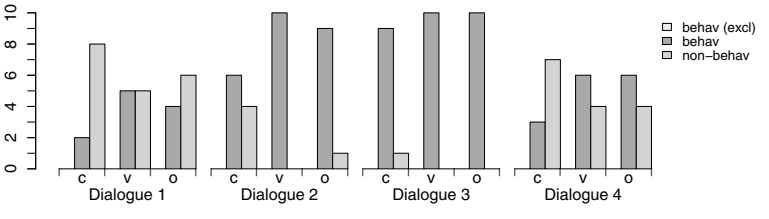


Fig. 2. Behaviour reports for the unrestricted set

When carrying out the same analysis in the restricted set, as illustrated on Figure 3, we once again verify that the amount of *behav (excl)* and *behav* summaries outnumber *non-behav* ones, to a considerable extent, in the second and third dialogues (63% and 90% respectively). By grouping these dialogues (*i.e.* the impolite) at one side and the first and fourth dialogues (neutral ones) at

[4] This problem was nevertheless found in only 69 of the 1,773 clauses, *i.e.* around 3.9% of them.

[5] Either 10% of the number of words in the source dialogue or no restriction at all.

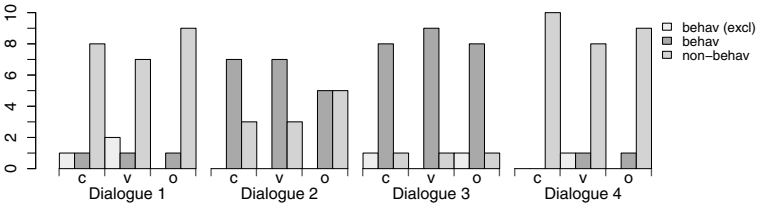


Fig. 3. Behaviour reports for the restricted set

the other, we can see that, even for summaries under the restricted condition, the number of behavioural and non-behavioural summaries depended on the politeness degree of the dialogue. This result was also statistically significant ($\chi^2(1, N = 120) = 45.95$, at the level $p = 0.001$). With these figures, we have confirmed hypothesis (i) and part of Hypothesis (iii) from [18].

Concerning hypothesis (ii), *i.e.* determining the existence of bias in the reports, bias was defined in [18] as a one-sided argument, "consisting of pure pro-argumentation for one side of an issue in a dialogue, while failing to genuinely interact with the other side on a balanced way" [22, pp. 86], *i.e.*, an argument failing to consider all relevant points from both sides. Thus, to determine the existence of bias in summaries, in [18] each summary was first classified either as an Exclusively Positive Report (EPR), *i.e.* a summary containing at least one sentence where some party was positively reported, with no negative reports on that party; or Exclusively Negative Report (ENR), when some party is reported negatively, without transfer of blame to some other agent, there also being no positive report about this party at all.

With our annotation scheme, these categories may be determined by taking together (i) *Type of Clause*, which is responsible for ruling *Neutral* clauses out of further consideration, since we are only interested in determining the presence of bias amongst emotional/behavioural reports; (ii) *Evaluated Party*, allowing us to identify towards whom the report was directed; and (iii) *Polarity*, so we can determine whether it was a positive or negative report. Hence, if, for example, all *IEmotional* clauses (as determined by *Type of Clause*) of some specific summary turn out to be Positive (at the *Polarity* dimension), with Vendor as the value for *Evaluated Party*, then this summary is taken to be an Exclusively Positive Report about the vendor[6].

Figure 4 shows the results for the unrestricted set of summaries, grouped by point of view. In this figure, it is clear the tendency customers have to report on the vendor's behaviour and vice-versa, *i.e.*, whose behaviour was negatively reported depended on the viewpoint assumed by the person reporting it ($\chi^2(2, N = 65) = 14.13$, at the significance level of $p = 0.001$), indicating the presence of a bias. When it comes to positive reports, however, no statistically significant relation between viewpoint and evaluated party was found ($\chi^2(2, N = 29) = 0.96$). Both results match those of [18].

[6] At this point, it is worth noticing that only the reliable and "tentative" dimensions were used in our analysis.

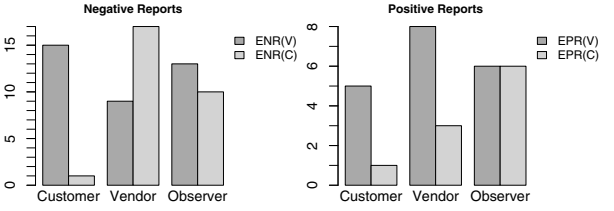


Fig. 4. Exclusively Positive and Negative Reports for the unrestricted set

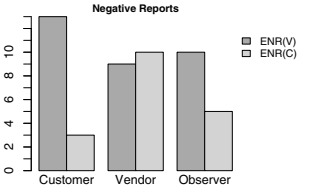


Fig. 5. Exclusively Negative Reports for the restricted set

As for the restricted set, figures change in that, contrary to what was predicted by [18], no relationship can be found between negatively reported behaviour and viewpoint ($\chi^2(2, N = 50) = 4.39$, at $p = 0.1$), as illustrated on Figure 5[7]. If, however, we only account for values under the customer and vendor viewpoints (and therefore put aside the neutral part, *i.e.* the observer), we end up with a statistically significant difference ($\chi^2(1, N = 35) = 4.27$, $p = 0.05$). The reason for this disparity might, in turn, rest on the low reliability of the *Consequence* dimension, which forced us to consider as ENR some summary that otherwise might not fall under this category, due to some undetected blame transfer. The fact that, in [18], the dataset was annotated by a single person, whereas in our research this annotation comes out from nine independent annotators, might have also played a part. Still, if we put all data together, and compare both unrestricted and restricted sets, we end up with 37 ENR(V) and 28 ENR(C) summaries, at the unrestricted side, and 32 ENR(V) and 18 ENR(C) at the restricted one, which is not significant at all ($\chi^2(1, N = 115) = 0.590, p > 0.25$), meaning that, as predicted in [18], considering a summary ENR(V) or ENR(C) did not depend on the summary length.

5 Conclusion

In this paper we presented a multi-dimensional annotation scheme for emotional and behavioural assessment in dialogue summaries. Reliability studies with nine independent annotators were carried out to test the soundness both of the annotation scheme and corresponding guidelines [2]. Results show that, from the

[7] In this figure, negative reports were left out because the dataset was too small to allow for any statistic analysis.

five original dimensions, three were reliable enough to allow for any conclusion to be drawn from the annotation, to wit, *Type of Clause*, *Evaluated Party* and *Polarity*. Unfortunately, *Intensity* and *Consequence* scored too low to deserve any credit.

To illustrate the utility of our scheme to other researchers, we have annotated data from an already published experiment (see [18]), verifying both (i) if it was possible to measure the same phenomena, and (ii) if the conclusions drawn with our annotation scheme would match those from that study. The comparison turned out to be successfull, in that our results confirmed all hypotheses setted up by [18]. We hope that, in using our scheme, researchers will be able to save a lot of time and effort that, otherwise, would be spent in planning, developing and testing a scheme of their own.

As for avenues for future work, we think it would be important determining a way to reliably measure intensity, since this is a feature considered in many theories about emotion (*e.g.* [17]). On this matter, simply adding intensity as a separate dimension definitely did not work for us. Finally, and as a measure to increase our confidence on the reproducibility of the scheme, it would be interesting having other volunteers apply it to a different dataset.

References

1. Alm, C.O., Roth, D., Sproat, R.: Emotions from text: Machine learning for text-based emotion prediction. In: Proceedings of HLT/EMNLP 2005, Vancouver, Canada (2005)
2. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. Computational Linguistics 34(4), 555–596 (2008)
3. Balahur, A., Lloret, E., Boldrini, E., Montoyo, A., Palomar, M., Martínez-Barco, P.: Summarizing threads in blogs using opinion polarity. In: Proceedings of the Events in Emerging Text Types Workshop of the RANLP, Borovets, Bulgaria (September 18, 2009)
4. Batliner, A., Huber, R., Niemann, H., Nöth, E., Spilker, J., Fischer, K.: The Recognition of Emotion. In: Foundations of Speech-to-Speech Translation, pp. 122–130. Springer, Heidelberg (2000)
5. Cohn, T., Callison-Burch, C., Lapata, M.: Constructing corpora for the development and evaluation of paraphrase systems. Computational Linguistics 34(4), 597–914 (2008)
6. Craggs, R., Wood, M.M.: A two dimensional annotation scheme for emotion in dialogue. In: AAAI Spring Symposium: Exploring Attitude and Affect in Text: Theories and Applications, Stanford, USA (March 2004); technical Report SS-04-07
7. van Deemter, K., Krenn, B., Piwek, P., Klesen, M., Schröder, M., Baumann, S.: Fully generated scripted dialogue for embodied agents. Artificial Intelligence 172(10) (June 2008)
8. Fischer, K.: Annotating emotional language data. Tech. Rep. 236, Verbmobil Project (December 1999)
9. Hijikata, Y., Ohno, H., Kusumura, Y., Nishida, S.: Social summarization of text feedback for online auctions and interactive presentation of the summary. Knowledge-Based Systems 20(6), 527–541 (2007)

10. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004), San Jose, USA, pp. 755–760 (July 2004)
11. Keenan, J., MacWhinney, B., Mayhew, D.: Pragmatics in memory: A study of natural conversation. Journal of Verbal Learning and Verbal Behavior 16(5), 549–560 (1977)
12. Krippendorff, K.: Content Analysis: An Introduction to its Methodology, 2nd edn. Sage, Thousand Oaks (2004)
13. Martin, J.R.: Beyond Exchange: Appraisal Systems in English. In: Evaluation in Text: Authorial Stance and the Construction of Discourse, pp. 142–175. Oxford University Press, Oxford (1999)
14. McKeown, K.R.: Discourse strategies for generating natural-language text. Artificial Intelligence 27(1), 1–41 (1985)
15. Miller, J.: An Introduction to English Syntax. Edinburgh University Press Ltd., Edinburgh (2002) ISBN 0 7486 1254 8
16. Mills, S.: Gender and Politeness. Cambridge University Press, Cambridge (2003)
17. Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press, Cambridge (1988)
18. Roman, N.T., Piwek, P., Carvalho, A.M.B.R.: Politeness and Bias in Dialogue Summarization: Two Exploratory Studies. In: Computing Attitude and Affect in Text: Theory and Applications. The Information Retrieval Series, vol. 20, pp. 171–185. Springer, Netherlands (January 9, 2006)
19. Spertus, E.: Smokey: Automatic recognition of hostile messages. In: Innovative Applications of Artificial Intelligence (IAAI 1997), pp. 1058–1065 (1997)
20. Turney, P., Littman, M.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems 21(4), 315–346 (2003)
21. Vieira, R., Poesio, M.: An empirically based system for processing definite descriptions. Computational Linguistics 26(4), 539–593 (2000)
22. Walton, D.: One-Sided Arguments: A Dialectical Analysis of Bias. State University of New York Press (1999)
23. Watts, R.: Politeness. Cambridge University Press, Cambridge (2003)
24. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. Computational Linguistics 35(3), 399–433 (2009)

Evolving Artificial Neural Networks Using Adaptive Differential Evolution

Adenilton J. da Silva, Nicole L. Mineu, and Teresa B. Ludermir

Center of Informatics, Federal University of Pernambuco,
Av. Luis Freire s/n CEP 50740-540, Cidade Universitária, Recife-PE, Brazil
{ajs3,nlm,tbl}@cin.ufpe.br

Abstract. One of the main problems in the training of artificial neural networks is to define their initial weights and architecture. The use of evolutionary algorithms (EAs) to optimize artificial neural networks has been largely used because the EAs can deal with large, non-differentiable, complex and multimodal spaces, and because they are good in finding the optimal region. In this paper we propose the use of Adaptive Differential Evolution (JADE), a new evolutionary algorithm based in the differential evolution (DE), to deal with this problem of training neural networks. Experiments were performed to evaluate the proposed method using machine learning benchmarks for classification problems.

Keywords: Artificial Neural Networks, Differential Evolution, JADE, Hybrid Systems.

1 Introduction

Differential evolution (DE) was proposed in 1995 by Storn and Price [1]. It is an effective evolutionary algorithm for numerical optimization of nonlinear multimodal problems and it has few control parameters. DE can be easily implemented and it has been successfully used in several benchmark problems and practical applications [2–5]. Despite the few number of control parameters, one of the main problems in DE is its parameter dependency [6]. Several papers proposed parameter adaptation mechanisms for DE [3, 7, 8].

In [3], some recent DEs are compared with the adaptive DE denominated JADE in the optimization of numerical benchmark problems. JADE has the control parameters changed in a process that uses feedback from the evolutionary search. JADE overcomes the other strategies in the majority of problems tested in [3].

DE algorithm has been applied with success to the training of neural networks models, for example for training feed-forward neural networks [9] optimizing the weights and architectures of a multi-layer perceptron [10] and to improve the performance of a fuzzy neural network [11].

Hybrid systems composed by evolutionary algorithms (EAs) and artificial neural networks (ANNs) have been proposed since 1990s. In [12] the author reviewed

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 396–405, 2010.

different combinations of EA and ANN and verified that this methodology helps in the search of "significantly better intelligent systems than relying on ANNs or EAs alone".

In this paper we propose the use of JADE to optimization of the architecture and the initial weights of a multilayer perceptron (MLP) neural network used for pattern recognition. The experiments performed with three classification databases from the PROBEN1 repository [13] and four training algorithms and DE, show that the use of JADE improves the performance of ANN.

The remainder of this paper is organized as follows: Section 2 presents DE and JADE algorithms. The proposed method is presented in section 3 and in the section 4 the experiments are described and the results are presented. Finally, in Section 5 the conclusions and future works are presented.

2 Evolutionary Algorithms

2.1 Differential Evolution

DE starts with a population of NP individuals of dimension D, where all individuals are initialized between some predefined boundaries. Each individual is a candidate solution for the objective function f to be minimized. In each generation g, DE mutates and crosses its population. The mutation step occurs as described in equation (1), where $\boldsymbol{x}_{basis,g}$ is the basis vector, $\boldsymbol{x}_{r_1,g}$ and $\boldsymbol{x}_{r_2,g}$ are two randomly chosen vectors and F is the scale factor.

$$\boldsymbol{v}_{i,g} = \boldsymbol{x}_{basis,g} + F(\boldsymbol{x}_{r_1,g} - \boldsymbol{x}_{r_2,g}) \tag{1}$$

The following step is the crossover, equation (2). In this step each member of the population $\boldsymbol{x}_{i,g}$ is crossed with a mutant vector $\boldsymbol{v}_{i,g}$. They exchange attributes in order to form the trial vector $\boldsymbol{u}_{i,g}$. The value j_{rand} is an attribute which ensures that $\boldsymbol{u}_{i,g}$ gets at least one component from $\boldsymbol{v}_{i,g}$.

$$\boldsymbol{u}_{j,i,g} = \begin{cases} \boldsymbol{v}_{j,i,g}, \texttt{if } rand_j(0,1) \leq CR \texttt{ or } j = j_{rand} \\ \boldsymbol{x}_{j,i,g}, \texttt{otherwise} \end{cases} \tag{2}$$

Then the trial vector $\boldsymbol{u}_{i,g}$ is evaluated and compared to the target vector $\boldsymbol{x}_{i,g}$. The member for the next generation, at $g = g + 1$, is described in equation (3), where the vector with the best fitness is selected to be $\boldsymbol{x}_{i,g+1}$. DE algorithm is described in algorithm 1.

$$\boldsymbol{x}_{i,g+1} = \begin{cases} \boldsymbol{u}_{i,g}, \texttt{if} f(\boldsymbol{u}_{i,g}) \leq f(\boldsymbol{x}_{i,g}) \\ \boldsymbol{x}_{i,g}, \texttt{otherwise} \end{cases} \tag{3}$$

There are different strategies for creating the mutant vector. The main difference among them is how to choose the basis vector and how many perturbations to use. In equation (4) and (5), the basis vector is randomly chosen, but in equation (4) there is only one perturbation while in equation (5) there are two

perturbations. In equation (6) there is one perturbation considering the best vector and the current vector and other perturbation of random vectors.

$$\texttt{DE/rand/1:}\boldsymbol{v}_{i,g} = \boldsymbol{x}_{r_0,g} + F(\boldsymbol{x}_{r_1,g} - \boldsymbol{x}_{r_2,g}) \tag{4}$$

$$\texttt{DE/rand/2:}\boldsymbol{v}_{i,g} = \boldsymbol{x}_{r_0,g} + F(\boldsymbol{x}_{r_1,g} - \boldsymbol{x}_{r_2,g}) + F(\boldsymbol{x}_{r_3,g} - \boldsymbol{x}_{r_4,g}) \tag{5}$$

$$\texttt{DE/current-to-best/1:}\boldsymbol{v}_{i,g} = \boldsymbol{x}_{i,g} + F(\boldsymbol{x}_{best,g} - \boldsymbol{x}_{i,g}) + F(\boldsymbol{x}_{r_0,g} - \boldsymbol{x}_{r_1,g}) \tag{6}$$

Algorithm 1. DE

```
1  begin
2      Create a random initial population
3      while termination criterion not met do
4          for i = 1 to NP do
5              Select basis vector x_{basis,g}
6              Randomly choose x_{r1,g} ≠ x_{basis,g} from current population P
7              Randomly choose x_{r2,g} ≠ x_{r1,g} ≠ x_{basis,g} from P
8              v_{i,g} = x_{basis,g} + F(x_{r1,g} − x_{r2,g})
9              Generate j_rand = randint(1,D)
10             for j = 1 to D do
11                 if j = j_rand or rand(0,1) < CR then
12                     u_{j,i,g} = v_{j,i,g}
13                 else
14                     u_{j,i,g} = x_{j,i,g}
15                 end
16             end
17             if f(x_{i,g}) ≤ f(u_{i,g}) then
18                 x_{i,g+1} = x_{i,g}
19             else
20                 x_{i,g+1} = u_{i,g}
21             end
22         end
23     end
24 end
```

2.2 JADE

JADE algorithm was proposed in [3] and the authors proposed a new mutation strategy denoted DE/current-to-pbest based in the DE/current-to-best, the use of an optional archive, and an adaptation parameter strategy.

The mutation strategy DE/current-to-pbest is described in equation (7), where p is in the interval $[0, 1)$, the value of $x^p_{best,g}$ is chosen from the $100p\%$ best individuals in the current generation, F_i is associated with x_i and it is regenerated at each generation, $x_{r1,g}$ is randomly selected from the current population and

$x_{r2,g}$ is randomly selected from the population (JADE without archive) or from the union of the population and one archive (JADE with archive).

$$\boldsymbol{v}_{i,g} = \boldsymbol{x}_{i,g} + F_i(\boldsymbol{x}^p_{best,g} - \boldsymbol{x}_{i,g}) + F_i(\boldsymbol{x}_{r1,g} - \tilde{\boldsymbol{x}}_{r2,g}) \tag{7}$$

Algorithm 2. JADE

```
begin
    Set μ_CR = 0.5; μ_F = 0.5; A = ∅
    Create a random initial population
    while termination criterion not met do
        S_F = ∅; S_CR = ∅
        for i = 1 to NP do
            Generate CR_i = randn_i(μ_CR, 0.1), F_i = randc_i(μ_F, 0.1)
            Randomly choose x^p_best,g as one of the 100p% best vectors
            Randomly choose x_r1,g ≠ x_i,g from current population P
            Randomly choose x_r2,g ≠ x_r1,g ≠ x_i,g from P ∪ A
            v_i,g = x_i,g + F_i(x^p_best,g − x_i,g) + F_i(x_r1,g − x_r2,g)
            Generate j_rand = randint(1, D)
            for j = 1 to D do
                if j = j_rand or rand(0, 1) < CR_i then
                    u_j,i,g = v_j,i,g
                else
                    u_j,i,g = x_j,i,g
                end
            end
            if f(x_i,g) ≤ f(u_i,g) then
                x_i,g+1 = x_i,g
            else
                x_i,g+1 = u_i,g; x_i,g → A;CR_i → S_CR; F_i → S_F
            end
        end
        Randomly remove solutions from A so that |A| ≤ NP
        μ_CR = (1 − c) · μ_CR + c · mean_A(S_CR)
        μ_F = (1 − c) · μ_F + c · mean_L(S_F)
    end
end
```

In the beginning of the algorithm the archive is empty and when a parent solution loses your position it is added to the archive. The archive has a limited size NP to avoid computational overhead. When the size of archive is bigger than NP one solution is randomly removed from the archive. The advantage in use the archive is to keep the diversity, increasing the amount of information available.

In JADE the parameters used in DE algorithm (CR and F) are calculated for each individual in the population. The value of CR_i is calculated using equation (8) and then truncated to $[0, 1]$, where $randn(\mu CR, 0.1)$ is a random number following normal distribution with mean μCR and standard deviation 0.1. The value of F_i is calculate using equation (9), where $randc(\mu F, 0.1)$ is a random number following Cauchy distribution with mean μF and standard deviation 0.1. If the generated value is bigger than 1, it will assume the value 1, and if it is lower than 0, it will be regenerated.

$$CR_i = randn(\mu CR, 0.1) \tag{8}$$

$$F_i = randc(\mu F, 0.1) \tag{9}$$

The values of μCr and μF are updated in each generation as described in equations (10) and (11). The values of S_{CR} and S_F are respectively the set of all successful crossover probabilities at the last generation and the set of all successful mutation factors at the last generation. JADE algorithm with archive is described in algorithm 2.

$$\mu CR = (1 - c) \cdot \mu CR + c \cdot mean_A(S_{CR}) \tag{10}$$

$$\mu F = (1 - c) \cdot \mu F + c \cdot mean_L(S_F) \tag{11}$$

3 Method

This section presents the proposed hybrid system. The main idea of the system is the use of evolutionary algorithms to optimize the neural network architecture and the initial weights. An individual encodes the information of learning algorithm and its parameters, architecture and weights of the network, and it is composed of the following parts: 'learning algorithm', 'learning algorithm parameters', 'layers', 'neurons', 'transfer function' and 'weights', as illustrated in Table 1.

Table 1. Composition of an individual

learning algorithm	learning algorithm parameters	layers	neurons	transfer function	weights

This representation uses only continuous values and the selection of each value for the parameters of the neural network is made considering the attribute with the most representative value. This encoding schema was also used in [13].

For the first part, 'learning algorithm', were considered four algorithms: Backpropagation (BP), Resilient Backpropagation (RB), Levenberg-Marquardt (LM) and Scaled Conjugate Gradient Backpropagation (SCG). BP is used to train the

ANN when the highest value is in the first attribute of the first part, RB is used when the highest value is in the second attribute and so on. If two or more attributes have the same value, it will be considered the attribute with the smallest index.

The second part represents the learning algorithm parameters and its size depends on the learning algorithms chosen to form the first part. All parameters of this part have real values, but they are not directly encoded. They are initialized between [-1.0, 1.0] and a linear map is used to obtain the real values of the parameters.

The third part encodes the number of layers of the ANN. In this paper the neural networks have a maximum of three hidden layers, so this part is a vector with three attributes. To determinate the number of hidden layers of the ANN is considered the attribute with the larger value. If the most representative value is in the nth coordinate, then the neural network will have n hidden layers.

The fourth part encodes the number of hidden neurons in each layer. In this paper the maximum number of neurons per layer is ten. The fifth part represents the transfer function of the neurons. For each layer there is a set of transfer functions that can be chosen: log-sigmoid, hyperbolic tangent sigmoid, linear and radial basis transfer function.

In the last part of the individual, the weights of the ANN are encoded. As they have continuous values, the weights are directly encoded on the individual and they are initialized in the range [-1.0, 1.0].

In the proposed hybrid system, the evolutionary algorithm is executed and the neural network is incorporated in the phase of evaluation of the individual. The information of each part of the individual is taken in order to form the neural network. After the structure is set, the neural network is trained with the evaluation and validation sets for 5 epochs, and for the fitness is considered the classification error over the evaluation set.

4 Experiments and Results

This paper proposes a new way to optimize the architectures and the initial weights of an artificial neural network. This optimization was realized through evolutionary algorithms JADE without archive (JADE) and JADE with archive (JADEa). In order to compare their performance, it was tested a neural network being optimized by DE and a neural network being trained through some classical training algorithms: Backpropagation, Resilient Backpropagation, Levenberg-Marquardt and Scaled Conjugate Gradient Backpropagation. We realize experiments with three well known classification problems from the PROBEN1 data set: card, cancer and diabetes.

There are several strategies for design experiments. Here we use the strategy presented in [14] because it is simple and highly used [13]. Following this strategy each data set is divided in two halves: one for train and the other for test. The half used for train is divided in two sets: the evaluation set (70% of the training data) and the validation set (with 30% of the training data).

The evolutionary algorithm uses only the evaluate set to calculate the fitness of the individuals, and then the validation set is used to verify the generalization capacity of the network. The EA has three outputs: 1) the network with the best fitness, denoted by the name of the evolutionary algorithm; 2) the network with the best validation error, denoted by the name of the evolutionary algorithm with the suffix val; and 3) the network (in the set of the networks with the best validation error in each generation) with the minimum value

$$0.3[(train\ error) - (validation\ error)] + 0.7(validation\ error), \tag{12}$$

denoted by the name of the evolutionary algorithm with the suffix trval.

In order to compare the obtained results, it was used the pared t-test with 5% of significance. All hybrid systems were compared to the ANN, and the comparison among hybrid systems were made considering the same type of error.

For each method the results are shown and the outputs of the same type are compared. The **boldface** line means that the method is better than the others with same output type. The *emphasized* lines mean that there is no difference between the *emphasized* method and the **boldfaced** method. Column HT from tables 2 to 4 shows if the method is statistically better than the ANN, where '=' means same performance, '<' means that the hybrid system overcomes the ANN.

Table 2 presents the results for Cancer problem. For type of error 'val' the hybrid systems DE current to best, JADE and JADEa are statistically better than ANN and there is no difference among them. For type of error 'trval' all hybrid systems are statistically better than ANN, but they are statistically equivalent among them.

Table 3 displays the results for Card problem, and for this problem DE current to best got the best value. The hybrid system DE current to best trval is

Table 2. Results for Cancer problem

	Train error		validation error		Test error		
Method	mean	std	mean	std	mean	std	HT
DE rand	0.0003	0.0010	0.0470	0.0187	0.0521	0.0128	=
DE current to best	0.0000	0.0000	0.0495	0.0189	0.0554	0.0116	=
JADE	0.0000	0.0000	0.0463	0.0209	0.0547	0.0163	=
JADEa	0.0001	0.0007	0.0451	0.0184	0.0503	0.0091	=
DE rand val	0.0297	0.0194	0.0159	0.0088	0.0481	0.0123	=
DE current to best val	0.0269	0.0181	0.0171	0.0098	0.0448	0.0111	<
JADE val	0.0321	0.0178	0.0171	0.0095	0.0455	0.0105	<
JADEa val	0.0280	0.0178	0.0171	0.0088	0.0460	0.0112	<
DE rand trval	0.0204	0.0118	0.0159	0.0088	0.0463	0.0091	<
DE current to best trval	0.0227	0.0131	0.0178	0.0096	0.0457	0.0088	<
JADE trval	0.0201	0.0093	0.0175	0.0097	0.0461	0.0097	<
JADEa trval	0.0205	0.0107	0.0171	0.0088	0.0458	0.0087	<
ANN	0.0035	0.0048	0.0435	0.0236	0.0549	0.0141	

Table 3. Results for Card problem

	Train error		validation error		Test error		
Method	mean	std	mean	std	mean	std	HT
DE rand	0.0365	0.0118	0.1793	0.0417	0.1868	0.0305	=
DE current to best	0.0241	0.0119	0.1796	0.0351	0.1816	0.0221	=
JADE	0.0260	0.0110	0.1819	0.0502	0.1845	0.0197	=
JADEa	0.0247	0.0087	0.1848	0.0388	0.1857	0.0252	=
DE rand val	0.1037	0.0284	0.1065	0.0230	0.1620	0.0219	<
DE current to best val	0.1096	0.0474	0.1061	0.0255	0.1628	0.0213	<
JADE val	0.1116	0.0413	0.1045	0.0260	0.1734	0.0262	=
JADEa val	0.0983	0.0376	0.1049	0.0249	0.1664	0.0207	=
DE rand trval	0.0968	0.0183	0.1078	0.0224	0.1657	0.0207	<
DE current to best trval	**0.1011**	**0.0263**	**0.1094**	**0.0261**	**0.1549**	**0.0181**	<
JADE trval	0.0972	0.0225	0.1071	0.0259	0.1678	0.0224	=
JADEa trval	*0.0934*	*0.0272*	*0.1078*	*0.0249*	*0.1605*	*0.0163*	<
ANN	0.0554	0.0128	0.1641	0.0407	0.1785	0.0274	

Table 4. Results for Diabetes problem

	Train error		validation error		Test error		
Method	mean	std	mean	std	mean	std	HT
DE rand	0.1167	0.0165	0.2603	0.0333	0.2732	0.0239	=
DE current to best	0.1071	0.0199	0.2586	0.0431	0.2759	0.0284	=
JADE	0.1144	0.0130	0.2733	0.0420	0.2780	0.0287	=
JADEa	0.1114	0.0148	0.2620	0.0403	0.2708	0.0197	=
DE rand val	0.1970	0.0392	0.1794	0.0271	0.2556	0.0191	<
DE current to best val	0.2019	0.0394	0.1777	0.0292	0.2564	0.0199	<
JADE val	**0.2035**	**0.0417**	**0.1800**	**0.0329**	**0.2457**	**0.0185**	<
JADEa val	*0.2066*	*0.0393*	*0.1820*	*0.0296*	*0.2526*	*0.0219*	<
DE rand trval	0.1839	0.0248	0.1832	0.0255	0.2530	0.0177	<
DE current to best trval	0.1853	0.0270	0.1820	0.0260	0.2532	0.0193	<
JADE trval	0.1887	0.0302	0.1843	0.0308	0.2511	0.0183	<
JADEa trval	0.1861	0.0234	0.1858	0.0288	0.2562	0.0241	<
ANN	0.1471	0.0206	0.2432	0.0355	0.2681	0.0215	

statistically better than ANN, but the hybrid system JADEa trval is statistically equivalent to it. The hybrid systems DE rand and DE current to best for type of error 'val' are statistically better than the ANN.

In Table 4 is shown the results for Diabetes problem. The hybrid system JADE Val is the best hybrid system and is statistically better than ANN. JADEa val is statistically equivalent to JADE val. For type of error 'trval' all hybrid systems are statistically equivalent and all of them are statistically better than ANN.

5 Conclusion

In this paper was proposed a hybrid system to train neural networks. The proposed system was compared with a neural network optimized by DE and some classical training algorithms. In the experiments we used three classification problems from the PROBEN1 data set.

For all data sets, the method that considers the best ANN as the one with the smallest classification error over the training set got overfitting and could not generalize well. This method does not improve the generalization capacity of the network. The methods that consider the validation error for selecting the best neural network got good results and almost all of them were statistically better than the artificial neural network.

For Cancer data set only the hybrid system DE rand val got the same performance of ANN. In Card data set, the hybrid system which stood out was DE current to best trval. DE current to best trval was the system with the smallest classification error and it was statistically better than the others systems of the same type, excepting JADEa.

In Diabetes data set, JADE and JADEa got the highlight. In this set, the best hybrid system was JADE val which was statistically better than the hybrid systems DE val.

The experiments showed that the neural networks optimized by the algorithms JADE and JADEa improved the performance of the ANNs. They also showed the importance of considering the validation error for the selection of the best neural network.

In future works to improve the generalization of the hybrid system one can use all members of the population to compose an ensemble instead to use the best individual, because the population can have more useful information than its best individual by itself [15]. To create the ensemble, niching techniques can be added to JADE in a way that the optimums found can be combined to compose the ensemble.

Acknowledgment

This work is supported by research grants from CNPq and FACEPE.

References

1. Storn, R., Price, K.: Differential evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute (March 1995)
2. Cai, H.R., Chung, C.Y., Wong, K.P.: Application of differential evolution algorithm for transient stability constrained optimal power flow. IEEE Transactions on Power Systems 23(2), 719–728 (2008)
3. Zhang, J., Sanderson, A.C.: Jade: Adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation 13(5), 945–958 (2009)

4. Kim, H., Chong, J., Park, K., Lowther, D.A.: Differential evolution strategy for constrained global optimization and application to practical engineering problems. IEEE Transactions on Magnetics 43(4), 1565–1568 (2007)
5. Das, S., Abraham, A., Konar, A.: Automatic clustering using an improved differential evolution algorithm. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 38(1), 218–237 (2008)
6. Gämperle, R., Müller, S.D., Koumoutsakos, P.: A parameter study for differential evolution. In: Proc. Advances Intell. Syst., Fuzzy Syst., Evol. Comput., pp. 293–298 (2002)
7. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: The 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1785–1791 (2005)
8. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation 10(6), 646–657 (2006)
9. Ilonen, J., Kamarainen, J., Lampinen, J.: Differential evolution training algorithm for feed-forward neural networks. Neural Processing Letters 17, 93–105 (2003)
10. Zarth, A., Ludermir, T.: Optimization of neural networks weights and architecture: A multimodal methodology. In: International Conference on Intelligent Systems Design and Applications, pp. 209–214 (2009)
11. Ma, M., Xu, Y., Zhang, L.: A method of improving performance of fuzzy neural network based on differential evolution. In: 2008 International Conference on Machine Learning and Cybernetics, vol. 2, pp. 874–877 (2008)
12. Yao, X.: Evolving artificial neural networks. Proceedings of the IEEE 87(9), 1423–1447 (1999)
13. Almeida, L., Ludermir, T.: A multi-objective memetic and hybrid methodology for optimizing the parameters and performance of artificial neural networks. Neurocomputing 73(7-9), 1438–1450 (2010)
14. Cantu-Paz, E., Kamath, C.: An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35(5), 915–927 (2005)
15. Liu, Y., Yao, X.: Evolving neural network ensembles by fitness sharing. In: IEEE Congress on Evolutionary Computation, CEC 2006, pp. 3289–3293 (2006)

A Recurrent Neural Network for Channel Assignment Problems in Mobiles

Gracián Triviño, José Muñoz, and Enrique Domínguez

E.T.S. Ingeniería Informática
Universidad de Málaga
{gracian,jose_munoz,enriqued}@lcc.uma.es

Abstract. The Channel Assignment Problem (CAP) is an NP-complete problem. Different algorithms have been proposed based on techniques of combinatorial optimization. In this paper, the problem is formulated as a minimizing problem of a quadratic function with quadratic restrictions and binary variables. To solve this problem a binary recurrent neural network (RNN) is proposed with a quadratic energy function. Constrains are incorporated to the computational dynamics so that RNN always satisfy the problem constraints in each updating. A variety of test problems are used to compare the performances of CAP against traditional heuristic approaches.

Keywords: channel assignment, neural network, combinatorial optimization.

1 Introduction

In the planning of radio mobile networks, operators need to assign frequencies to the radio stations minimizing the possibilities of interference. The channel assignment problem consists of assigning M channels to N cells taking into account that a channel can be shared by various non-contiguous cells and contiguous channels cannot be assigned to the same station. The fact that the electromagnetic spectrum available for the purpose is a limited resource places severe limitations on the size and performance of such system. So, a number of previously fixed channels, T_i, are assigned to each station i, i=1, 2,…, N.

The increase of the number of channels and cells makes it difficult to find a solution to the problem since CAP is a *NP*-complete problem [1] since is a generalized graph-coloring problem. In literature a great number of heuristic algorithms have been proposed to solve the problem [1-6]. Among the disadvantages of this methods we can point out that they are sequential and need parameter fitting.

On the other hand, other types of algorithms have been suggested based on neural networks [7-13], which are different in their mathematical formulations, the computational dynamics of the neural network and the mechanism to escape from local minima. These algorithms assure the convergence forward local minima since the energy function since the dynamical rule can only decrease the energy function.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 406–412, 2010.

Among their inconveniences we can point out that they are sequential and require usually parameter fitting. Moreover, they do not find the ideal solution of some tests used in literature [13].

The paper is organized as follows. In section 2, a new formulation for the CAP is proposed. In section 3 we design a new neural approach based on synchronous (parallel) updating and without parameter fitting. An improved approach to escape from the local minima is presented in section 4. Experimental results comparing the performance of the simulated neural approach and other classical methods are provided in section 5. Finally, conclusions are presented in the last section 6.

2 Problem Formulation

Let $V_{i,j}$ be the state variable associated to the neuron (i,j).

$$V_{ij} = \begin{cases} 1 & \text{if the chanel } j \text{ is asigned to cell } i \\ 0 & \text{otherwise} \end{cases}$$

The adjacent matrix A=(a_{ih})) is used to measure the degree of interference between cell i and h. Thus,

$$a_{ih} = \begin{cases} 1 & \text{if the chanel } i \text{ is adjacent to cell } h \\ 0 & \text{otherwise} \end{cases}$$

The CAP is defined by the following constrains:

$$\sum_{j=1}^{M} V_{ij} = T_i, \quad i = 1,2,...,N \tag{1}$$

That is, T_i channels have to be assigned to cell i. Moreover,

$$\sum_{k=1}^{M}\sum_{h=1}^{N}\sum_{i=1}^{N} a_{ih} * V_{ik} * V_{hk} = 0 \tag{2}$$

since two contiguous cells cannot share the same channel. Finally,

$$\sum_{i=1}^{N}\sum_{j=1}^{M-1} V_{ij} * V_{i(j+1)} = 0 \tag{3}$$

That is, two adjacent channels can not be assigned to the same cell.

A solution for set of constraint can be found by minimization of the following function:

$$E = \sum_{i=1}^{N}(\sum_{j=1}^{M} V_{ij} - T_i)^2 + \sum_{k=1}^{M}\sum_{h=1}^{N}\sum_{i=1}^{N} a_{ih} * V_{ik} * V_{hk} + \sum_{i=1}^{N}\sum_{j=1}^{M-1} V_{ij} * V_{i(j+1)} \tag{4}$$

3 Neural Representation

To find a local minimum of problem (4) can be used a Hopfield binary neural network. By identification of terms with respect to the Hopfield energy function,

$$E = -1/2 \sum_{i=1}^{N} \sum_{j=1}^{N} W_{ij} * V_i * V_j + \sum_{i=1}^{N} \theta_i * V_i \tag{5}$$

we obtain the synaptic weights $w_{ij,hk}$ and the threshold values θ_{ij}:

$$w_{ij,hk} = \begin{cases} -2 & \text{if } i = h \text{ y j} \neq 1 \\ -2a_{ih} & \text{if } j = k \text{ y } i \neq h \\ -2 & \text{if } j = k-1 \text{ y } i = h \\ -2 & \text{if } j-1 = k \text{ y } i = h \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$\theta_{ij} = -2T_i + 1 + a_{ii}$$

4 A New Neural Model

Neural networks have been used to solve many various problems in the telecommunication area [11-13]. Hopfield have also been used to solve the CAP with mixed success [7-10]. In this section a Hopfield network approach is introduced. The proposed neural approach is able to ensure feasibility of the final solution and incorporates an internal dynamics to escape from local minima. Feasibility problems and poor solution quality in the Hopfield networks can be eliminated by an appropriate form of the energy function and modification of the computational dynamics. We consider the objective function of the CAP (7) that differs of the expression (4) in the first term. This term is not necessary since the new computational dynamics provides also solutions which satisfy this constrain. That is, the number of active neuron in the row i, (i,1), (i,2),…,(i, M), is always equal to T_i.

$$E = \sum_{k=1}^{M} \sum_{h=1}^{N} \sum_{i=1}^{N} a_{ih} * V_{ik} * V_{hk} + \sum_{i=1}^{N} \sum_{j=1}^{M-1} V_{ij} * V_{i(j+1)} \tag{7}$$

The synaptic weight matrix $w_{ij,hk}$ and threshold θ_{ij} of the proposed neural network are the following:

$$w_{ij,hk} = \begin{cases} -2a_{i,h} & \text{if } j = k \text{ y } i \neq h \\ -2 & \text{if } j = k-1 \text{ y } i = h \\ -2 & \text{if } j-1 = k \text{ y } i = h \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$\theta_{ij} = 0, \ \forall i, j$$

Thus, the neural architecture is organized as a matrix where each neuron V_{ik} is only connected to the neurons situated at its right and left ($V_{i(k+1)}$ and $V_{i(k-1)}$) and all neurons in the same column (V_{ij}, j=1,2,…,M), as shown in figure 1.

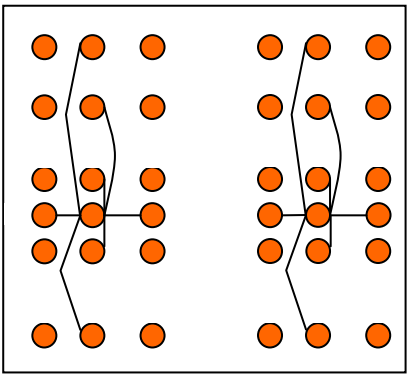


Fig. 1. Neural architecture

The activation potential of the neuron (*i, j*) is given by

$$H_{V_{ij}} = \sum_{h=1}^{N} a_{ih} V_{ij} V_{hj} + V_{ij} V_{i(j-1)} + V_{ij} V_{i(j+1)}, \qquad i = 1,2,...,N,\ j = 2,...,M-1 \tag{9}$$

which can be identified as the contribution of the neuron V_{ij} to the energy function.

To satisfy the *N* restrictions

$$\sum_{j=1}^{M} (V_{ij} - T_i)^2 = 0, \quad i = 1,2,...,N$$

the computational dynamics of the network is conveniently modified so that always the same amount of neurons in each row are fired. That is, T_i neurons in row *i*, *i*=1,2….,*N*. A possible way to maintain this objective is to find two neurons with contrary value which can be exchanged so that the network energy is not increased. Therefore we must choose randomly two neurons from the same row *i* which different states. This is, two neurons V_{ij} and V_{ik} so that $V_{ij} \neq V_{kl}$.

The contribution of the neurons V_{ij} and V_{ik}, to energy of the network is given by the following expression:

$$H_{V_{ij}V_{ik}} = \sum_{r=1}^{N} a_{ir} (V_{ij} V_{rj} + V_{ik} V_{rk}) + V_{ij} V_{i(j+1)} + V_{i(j-1)} V_{ij} + V_{ik} V_{i(k+1)} + V_{i(k-1)} V_{ik}$$
$$i = 1,2,...,N,\ j = 2,...,M-1,\ r = 1,2,...,N \tag{10}$$

If we choose V_{ij} and V_{ik} so that $V_{ij} \neq V_{ik}$ and assume that V_{ij}=1 and V_{ik} =0, we obtain the expression:

$$H_{V_{ij}=1/V_{ik}=0} = \sum_{r=1}^{N} a_{ir} V_{rj} + V_{i(j+1)} + V_{i(j-1)} \tag{11}$$

which represents the contribution of the neurons V_{ij} and V_{ik} to the network energy, when V_{ij} =1 and V_{ik}=0. Similarly, we have

$$H_{V_{ij}=0/V_{ik}=1} = \sum_{r=1}^{N} a_{ir} V_{rk} + V_{i(k+1)} + V_{i(k-1)} \tag{12}$$

Therefore we can exchange the values of the neurons if the following condition is satisfied:

$$H_{V_{ij}=1/V_{ik}=0} - H_{V_{ij}=0/V_{ik}=1} > 0 \tag{13}$$

It guarantees that the network energy decrease and the number of channels has not be modified in the updating. Therefore the computational dynamics consists on choosing two neurons from a row so that they have different values and that the condition (13) be satisfied. Each neuron is updated with its complementary value.

5 Stochastic Dynamics

To try to escape from local minima the condition (13) is modified in the following way:

$$H_{V_{ij}=1/V_{ik}=0} - H_{V_{ij}=0/V_{ik}=1} > \sigma \cdot randn \cdot E_k / k \tag{14}$$

where k is the actual iteration; E_k is the value of the energy function in the k-esima iteration, *randn* is a value of a Gaussian distribution with mean 0 and standard deviation 1 and the parameter σ is the level of improvement. If the right term in (14) is positive and k is small then the new solution is better that old solution and the

Table 1. Average values of the energy function

Problem	HN	HCHN	SONN	RNN
HEX1	49.0	48.7	53.0	24
HEX2	21.2	19.8	28.5	0
HEX3	81.6	80.3	87.2	24
HEX4	21.6	18.9	29.1	0
KUNZ1	22.1	21.1	22.0	40
KUNZ2	32.8	31.5	33.4	62
KUNZ3	13.2	13.0	14.4	26
KUNZ4	0.4	0.1	2.2	0

energy function is decreased according to the right term from (14). On the other hand, if the right term is negative then the new solution can be worse than the old one but without overcoming certain limits.

6 Experimental Results

The results presented in Table 1 show the average values of the energy function of three conventional heuristics applied to CAP (HN, HCHN and SONN described in [7]) with the proposed neural network (RNN) in this paper.

Can be seen in table 1 that proposed RNN has achieved an optimal solution in three problems. Note that the rest of the algorithms do not achieved an optimal solution for theses problems. In figure 2 is shown the energy function of RNN network for the problem KUNZ4 that has a strong decreasing of the energy function in the initial iterations.

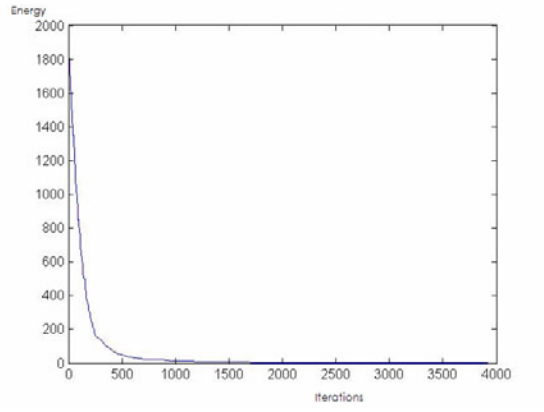


Fig. 2. Evolution of the energy for KUNZ4

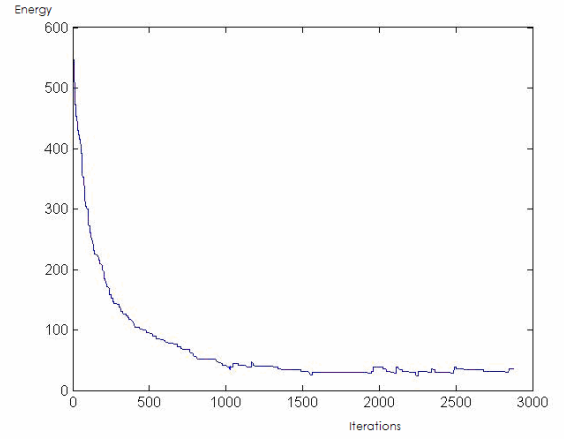


Fig. 3. Evolution of the energy function in HEX1

On the other hand, in figure 3 you can see the curve of the energy function for the HEX1 problem which has small increasing when el number of iterations is greater than a thousand. So, the algorithm can escape from local minima.

We have not found an optimal solution for HEX1 problem.

7 Conclusions

In this paper a neural algorithm is proposed for the channel assignment problem to mobile cells which allow its parallel implementation, since it is based on the exchange of values in pairs of neurons with different states which can update simultaneously with pairs of neurons that are not in the same row nor are adjacent. The computational dynamics always guarantees the decrease of the energy function except when a random noise is inserted to escape from a local minimum. Likewise, it is a recurrent network which is free of parametric adjustment; it is computing efficient and finds optimal solutions in problems where other algorithms have not achieved optimal solution.

References

1. Hale, W.K.: Frequency assignment: theory and applications. Proc. IEEE 68, 1497–1514 (1980)
2. Pennoti, R.J.: Channel assignment in cellular mobile telecommunication systems. M.S. thesis. Polytechnic Inst. of New York, Brooklyn (1976)
3. Box, F.: A heuristic technique for assignment frequencies to mobile radio nets. IEEE Trans. Veh. Technol. VT-27, 57–64 (1977)
4. Gamst, A., Rave, W.: On frecuency assignment in mobile automatic telephone systems. In: Proc. GLOBECOM 1982, pp. 309–315 (1982)
5. Sivarajan, K.N., McEliece, R.J., Ketchum, J.W.: Dynamic channel assignment in cellular radio. In: Proc. IEEE Vehicular Technology Conf. (VTC), pp. 631–635 (1990)
6. Sivarajan, K.N., McEliece, R.J., Ketchum, J.W.: Channel assignment in cellular radio. In: Proc. IEEE Vehicular Technology Conf. (VTC), pp. 920–924 (May 1989)
7. Smith, K., Palaniswami, M.: Static and Dynamic Channel Assignment Using Neural Networks. IEEE Journal on Selected Areas of Communications 15(2) (February 1997)
8. Zabalawi, I., Jaradat, A., Al-Khawaldeh, R.: A dinamic channel assigment technique based on the discrete Hopfield neural network model. In: Fifth International Symposium on Signal Processing and its Applications (August 1999)
9. del Re, E., Fantacci, R., Ronga, L.: A Dynamic Channel Allocation Technique Based on Hopfield neural networks. IEEE Transactions on Vehicular Technology 45(1) (February 1996)
10. Wang, J., Tang, Z., Xu, X., Li, Y.: A discrete competitive Hopfield neural network for cellular channel assignment problems. Neurocomputing 67, 436–442 (2005)
11. Chan, P.T.H., Palaniswami, M., Everitt, D.: Neural Network-based dynamic channel Assignment for cellular mobile commutation systems. IEEE Transactions on Vehicular Technology 42(2) (May 1994)
12. Funabiki, N., Takefuji, Y.: A neuronal network parallel algorihm for channel assignment problems in cellular radio networks. IEEE Transactions on Vehicular Technology 41(4) (November 1992)
13. Kunz, D.: Channel assignment for cellular radio using neural networks. IEEE Transactions on Vehicular Technology 40(1) (February 1991)

SVM Based Feature Selection: Why Are We Using the Dual?

Guillermo L. Grinblat, Javier Izetta, and Pablo M. Granitto*

CIFASIS, French Argentine International Center for Information and Systems Sciences, UPCAM (France) / UNR–CONICET (Argentina),
Bv. 27 de Febrero 210 Bis, 2000 Rosario, Argentina
{grinblat,izetta,granitto}@cifasis-conicet.gov.ar

Abstract. Most Support Vector Machines (SVM) implementations are based on solving the dual optimization problem. Of course, feature selection algorithms based on SVM are not different and, in particular, the most used method in the area, Guyon et al.'s Recursive Feature Elimination (SVM-RFE) is also based on the dual problem. However, this is just one of the options available to find a solution to the original SVM optimization problem. In this work we discuss some potential problems that arise when ranking features with the dual-based version of SVM-RFE and propose a primal-based version of this well-known method, PSVM-RFE. We show that our new method is able to produce a better detection of relevant features, in particular in situations involving non-linear decision boundaries. Using several artificial and real-world datasets we compare both versions of SVM-RFE, finding that PSVM-RFE is preferable in most situations.

1 Introduction

Several current problems of high technological importance in machine learning, as for example text processing, gene expression analysis or in-silico chemistry, share the characteristic of presenting much more features than measured samples [11]. In most cases several of these variables have a relatively low importance for the problem at hand. In fact, they usually interfere with the learning process instead of helping it, a problem usually known as "the curse of dimensionality".

Feature selection techniques aim at the solution of this problem[14]. Their main objective is to find a small subset of the measured variables that improve, or at least do not degrade, the performance of the modeling method applied to the dataset. The application of feature selection methods is beneficial in different ways. Not only they avoid the curse of dimensionality, but also they allow a considerable reduction in complexity of the models and an easier visualization and interpretation of the data under analysis.

The Support Vector Machine (SVM, [5,18]) is probably the most used method in machine learning nowadays. It has many interesting qualities, among which

* Work supported by ANPCyT grants 643 and 2226.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 413–422, 2010.

outstand a solid theoretical foundation and the simple requirement of solving a quadratic optimization problem (with unique solution). The method was originally developed as a simple linear classifier, but a powerful extension to non-linear problems, based on the use of kernels [17], was later introduced [2]. Of course, several methods for feature selection based on SVM where developed in the last years [19,9]. The most relevant is clearly Guyon et al.'s Recursive Feature Elimination (SVM-RFE, [12]). The focus of Guyon et al.'s work is on feature selection in the linear case, but they briefly discuss an extension to the use of non-linear kernels. Some other works also introduced improvements to the basic SVM-RFE method. For example, Rakotomamonjy [16] discuss the use of the weight vector derivative as a ranking measure. Duan et al. [6] use a statistical analysis of weight vectors from multiple linear SVMs trained on subsamples of the original training data, instead of the weight vector obtained directly using all the data. Finally, Zhou et al. [21] presented several extensions to multiclass problems.

The minimization problem presented in SVM is usually solved by means of its dual problem. Only a few works considered the problem of directly solving the SVM in its primal version [13,15]. Recently, in a very interesting paper Chapelle [4] introduced a solution to this problem, even in the more difficult non-linear case. The author argues that using the primal or the dual are in fact equally complex ways of solving SVM's optimization problem. On the other hand, if the objective is to find only an approximate solution to the SVM problem, Chapelle claims that the primal version is to be preferred because there are no guaranties that an approximate solution of the dual problem is in fact an approximate solution of the primal problem, which is the one that needs to be solved.

SVM-RFE iteratively eliminates the less relevant features according to a ranking based on the dual SVM problem. As a consequence, variables with some particular characteristics are not ranked in the right position, leading in some cases –as we will show later in this paper– to a bad general performance of the method. In this paper we propose to select features in SVM-RFE using a ranking based directly in the solution of the primal problem. We show a simple method to implement it, leading to better selection results in most cases.

In the next section we briefly review both primal and dual SVM problems and the original version of SVM-RFE. In Section 3 we discuss the limitations of the dual SVM-RFE with a motivational example, followed in Section 4 with the introduction of our new version of SVM-RFE. Finally, in Section 5 we show a series of evaluations of the method and close in Section 6 with some concluding remarks.

2 SVM in Its Primal and Dual Form

In this section we briefly review both the primal and dual formulation of the SVM problem. We also recall SVM-RFE, the most used algorithm for feature selection with SVM.

2.1 SVM Formulation

Given a dataset $[(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)]$, where $\mathbf{x}_i$ is a vector in a given vector space and $y_i = \pm 1$, SVM searches for the separating hyperplane ($y = \mathbf{w}\mathbf{x} + b$) that maximizes the margin (the smallest distance between the hyperplane and any point in the dataset). When a non-linear decision surface is needed, the method uses a mapping Φ from the original vector space to another space of a higher dimension –even infinite– and then searches for the optimal hyperplane in the new vector space. A useful property of SVMs is that there is no need to know the non-linear mapping explicitly, it is enough to have a *kernel* function, k, that evaluates the inner product of two points in the mapped space.

The problem is formulated as

$$\begin{aligned} \min \ & ||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i, \\ \text{s.t.} \ & \xi_i \geq 0 \\ & y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) - 1 + \xi_i \geq 0. \end{aligned} \tag{1}$$

where C is a constant and the ξ_i are the so-called slack variables. This formulation is known as the primal problem. Usually, the SVM problem is solved using the dual formulation:

$$\begin{aligned} \max \ & -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{n} \alpha_i, \\ \text{s.t.} \ & \sum_{i=1}^{n} y_i \alpha_i = 0, \ 0 \leq \alpha_i \leq C, \end{aligned} \tag{2}$$

where the α are the Lagrange coefficients. At the optimal value, $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x}_i)$.

As explained by Chapelle [4], the change from solving the primal to the dual is usually based in two facts: i) it is easier to deal with the constraints efficiently and ii) kernels can be incorporated in the dual formulation without much effort. However, Chapelle also highlights in his work that there are no real reasons that prevent the direct solution of the primal problem. In fact, problem (1) can be posed as an unconstrained problem:

$$\min \ ||f||_{\mathcal{H}}^2 + C \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i) + b), \tag{3}$$

where f lies in the Reproducing Kernel Hilbert Space $\mathcal{H}$ associated to k and $L(y, x) = \max(0, 1 - yx)$, is the *hinge loss*. It is worth mentioning at this point that the use of this particular loss function is not critical to the method and can be easily replaced by other loss function with better properties.

The Representer Theorem implies that $f(\mathbf{x}) = \sum_{i=1}^{n} \beta_i k(\mathbf{x}_i, \mathbf{x})$, which means that (3) can be rewritten as

$$\min \sum_{i,j=1}^{n} \beta_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j) + C \sum_{i=1}^{n} L(y_i, \sum_{j=1}^{n} \beta_j k(\mathbf{x}_j, \mathbf{x}_i) + b). \tag{4}$$

If we use a differentiable L function, the problem can be solved by a simple gradient descent method. Chapelle shows an example using a Newton method for a particular differentiable approximation of the hinge loss. In the limit, when the approximation tends to the hinge loss, $\beta_i = y_i \alpha_i$.

2.2 SVM-RFE

The SVM-RFE selection method [12] is a recursive process that ranks variables according to a measure of their importance based on SVMs. At each iteration the importance of a set of variables is measured and the less relevant one is removed. Another possibility, which is the most commonly used, is to remove a group of features each time, in order to speed up the process. Usually, 10% of the variables are removed at each step until the number of variables reaches a lower limit, and from that point on the variables are removed one at a time [8]. The recursion is needed because for SVMs the relative importance of each feature can change substantially when evaluated over a different subset of features during the stepwise elimination process –in particular for highly correlated features–. The –inverse– order in which features are eliminated is used to construct a final ranking. The feature selection process itself consists only in taking the first n features from this ranking.

The key of the method is the measure used to rank the features. Guyon et al. start by fitting an SVM to the training data. Then, they consider the cost function of the dual problem (2):

$$J = \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{n} \alpha_i$$

and, in particular, the change in J when variable v is eliminated from the problem, $DJ(v)$. The authors argued that the importance of a variable is directly correlated with $DJ(v)$. In order to limit the computational burden, it is assumed that the vector α does not change with this elimination. According to this, the importance of a variable is measured by:

$$DJ(v) = \left| \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(k(\mathbf{x}_i, \mathbf{x}_j) - k_{-v}(\mathbf{x}_i, \mathbf{x}_j) \right) \right|, \tag{5}$$

where k_{-v} is the value of the kernel function computed excluding variable v. It is easy to see that in the case of a linear kernel the importance measure is reduced to the component of $\mathbf{w}$ in the corresponding direction [12].

3 The Problem with Dual-Based SVM-RFE

From the definition of $DJ(v)$ it is possible to deduce that the change in a given direction given by $k(\mathbf{x}_i, \mathbf{x}_j)$ has opposite contributions to $DJ(v)$ when x_i and x_j belong to the same class ($y_i \cdot y_j = 1$) or when they belong to different classes ($y_i \cdot y_j = -1$). In the particular case of the Gaussian kernel [17], the term $k(\mathbf{x}_i, \mathbf{x}_j) - k_{-v}(\mathbf{x}_i, \mathbf{x}_j)$ is always positive: k is a monotonous function of the distance $||x_i - x_j||$ and this distance never increases when a dimension is eliminated. It follows that, for the Gaussian kernel, the approach of some support vectors belonging to different classes produced by the elimination of a given variable can be balanced with the approach of support vectors belonging to the same class. A given variable for which both effects are similar will have a low ranking, independently of its real importance for the classification problem.

This situation can be better explained with a simple example. We consider a dataset comprising two classes, with a total of 100 variables but only two of them (x_1 and x_2) relevant to the separation of the classes. x_1 is sampled from a centered, normal distribution with $\sigma = 0.05$ for one class and from an uniform distribution between -1 and 1 for the other class. x_2 is the opposite, sampled from a normal distribution ($\mu = 0, \sigma = 0.05$) for the second class and from a uniform distribution ($[-1 : 1]$) for the first class. All other 98 irrelevant variables $x_3, \ldots, x_{100}$ are sampled from a normal, centered distribution with $\sigma = 1$. Left panel of Figure 1 shows the projection of the dataset over x_1 and x_2.

The optimal decision surface for this problem is non-linear. Over the plane x_1, x_2, the Bayes decision surface is formed by the lines $x_1 = x_2$ and $x_1 = -x_2$. Both variables are equally relevant and necessary for the optimal solution. When considering the 100 dimensions, the solution of the SVM with a Gaussian kernel selects nearly all points as support vectors. Given that, both variables receive a very low ranking from the dual SVM-RFE.

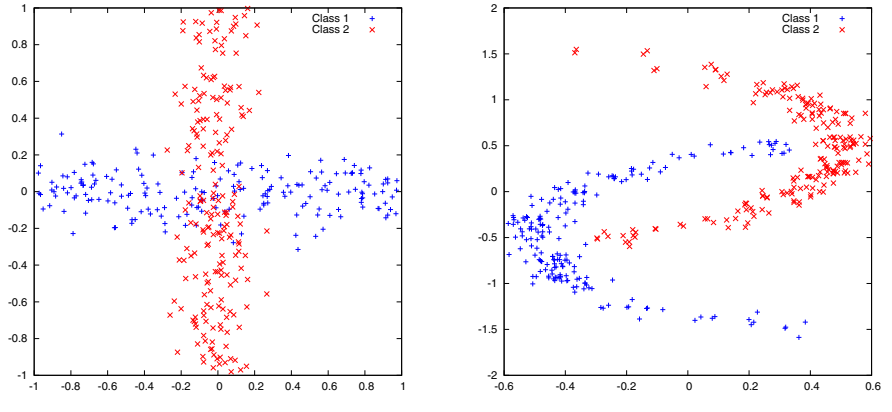


Fig. 1. Two examples of artificial datasets for which dual feature selection is not effective

4 Primal SVM-RFE

In order to overcome the described potential drawback of SVM-RFE in non-linear cases, we propose here to rank the variables using directly the change in the cost function of the primal problem. Starting now from equation (4), and considering that $\beta_i = y_i \alpha_i$, we can define a new importance for each variable given by:

$$DJ_p(v) = \left| \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(k(\mathbf{x}_i, \mathbf{x}_j) - k_{-v}(\mathbf{x}_i, \mathbf{x}_j) \right) + \right.$$
$$\left. C \sum_{i=1}^{n} (L(y_i, \sum_{j=1}^{n} \alpha_j y_j k(\mathbf{x}_j, \mathbf{x}_i) + b) - L(y_i, \sum_{j=1}^{n} \alpha_j y_j k_{-v}(\mathbf{x}_j, \mathbf{x}_i) + b)) \right|.$$

As in the dual case, we considered that the support vectors do not change when a variable is removed. It is important to note here that if we compute the real change in the cost function (finding the solution for the SVM problem after eliminating each variable) there will be no differences between the primal and dual solutions and the ranking of variables will be the same. The real change between both versions comes from the use of an approximation to the value of the difference (the vector of α does not change). Our new primal version (PSVM-RFE from here on), that takes into account directly the error rate in the solution, can lead to better rankings in non-linear cases, as we confirm in the next section with some examples.

5 Empirical Evaluation

We tested PSVM-RFE using several artificial and real-world datasets. In all the experiments reported in this work we followed the same procedure. We considered SVM with a Gaussian kernel. We used a 5 fold cross-validation procedure to select the optimum values of C and γ for the complete dataset (all variables). Then, we computed $k = \gamma/\overline{\gamma}$, where $\overline{\gamma}$ is the mean inter-point distance in the dataset. $\overline{\gamma}$ is usually used directly as a good approximation to the optimal γ [3]. Each time we eliminated a variable from a dataset we updated γ, defining it as $k\overline{\gamma}_i$, where $\overline{\gamma}_i$ is the new mean inter-point distance in the dataset, considering only the variables that remain in the problem. This is a better approximation to the optimal γ, in which we are assuming that the ratio between the optimal value and $\overline{\gamma}$ does not change. We used exactly the same procedure for both methods under comparison, the original SVM-RFE and the new PSVM-RFE.

5.1 Artificial Datasets

We used two artificial datasets to check the new method in the difficult situations for feature selection described in Section 3. The first dataset was already introduced in that section. The second one is the well-known Two moons dataset. Each class

Table 1. Test results on the Artificial datasets. The two columns show how many times (over the 30 realizations) each method positioned the two informative variables (column "Two variables"), or at least one of them (column "One variable"), into the first two positions of the ranking.

		Two variables	One variable
Dataset 1	SVM-RFE	0	28
	PSVM-RFE	30	30
Dataset 2	SVM-RFE	23	30
	PSVM-RFE	29	30

has a curved shape, with a higher density at the center of the class. The right panel of Figure 1 shows a realization of the dataset, considering only the two discriminant variables in the problem. As in the first case, we added 98 non-informative variables sampled from a normal distribution. In both datasets we created two separated sets for training and testing, each one with 200 datapoints.

In Table 1 we show the results of the experiments with both feature selection methods. The two columns show how many times (over 30 realizations) each method ranked the two informative variables, or one of them, into the first two positions. It is clear from the table that both datasets are difficult for SVM-RFE, but our new PSVM-RFE can deal efficiently with them.

5.2 Real World Datasets

We also compared both versions of RFE on three real-world datasets with different characteristics.

- *Qsar*. This dataset was used in a previous study on SVM-RFE for the prediction of pharmacokinetic and toxicological properties of chemical agents [20]. In this case the task is to predict which composts are the P-glycoprotein (P-gp) substrates, based on molecular descriptors. P-gp substrates were collected from the literature that are either described as being transported by P-gp or reported to induce overexpression of P-gp. Nonsubstrates of P-gp are those specifically described as not transportable by P-gp. A total of 116 substrates and 85 nonsubstrates of P-gp were collected. A total of 159 molecular descriptors were selected for the prediction task.
- *Nostrani*. The second dataset, previously used in [10], comprises a sensory evaluation of six typical "Nostrani" cheeses interesting as possible candidates of PDOs (Protected Designation of Origin). All of them are made with raw bovine milk, in six different cheese factories located at Trentino area, Italy. Sixty samples, 10 for each product, have been collected and evaluated by a panel of 8 assessors using 35 attributes to describe each sample. As a result, the dataset contains 60 samples, 10 for each of the six classes, and 35 features.
- *Optdigits*. The last one is the Optdigits dataset, available at the UCI Repository [7]. This dataset has 5620 instances with 64 features that characterize

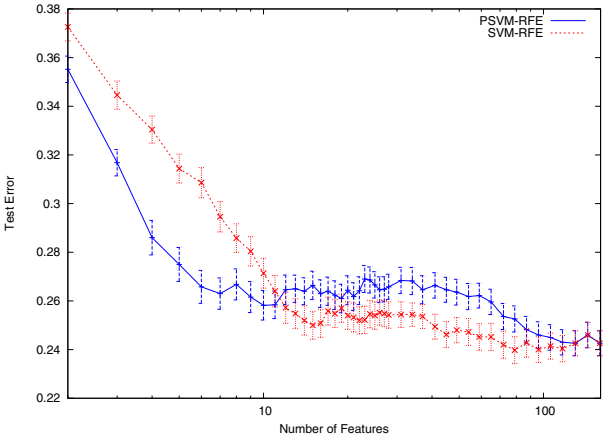


Fig. 2. Direct comparison of SVM-RFE and PSAM-RFE on the Qsar dataset. The figure shows mean values over 100 experiments. The error of the mean is also showed.

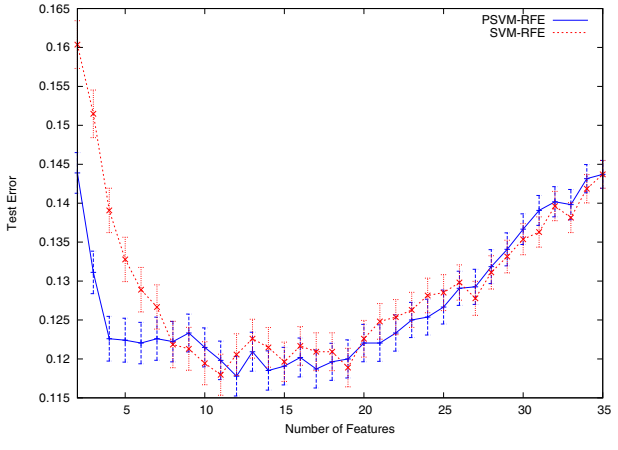


Fig. 3. Same as Figure 2 for the Nostrani dataset

32x32 bitmaps of handwritten digits from 43 individuals [1]. We selected the task of separating digit '8' from the rest.

For the Qsar and Nostrani datasets we performed 300 experiments as described next, and 100 for the longer Optdigits dataset. In each case we first randomly selected a test set (50 datapoints for Qsar, 18 for Nostrani and 400 for Optdigits). We used all other samples for training in the first two datasets and another random sample with 400 datapoints in the Optdigits case. For the selection of the hyper-parameters (C and γ) we used the same procedure as with the artificial datasets. For the Nostrani and Optdigits we eliminated one feature at a time, but for the

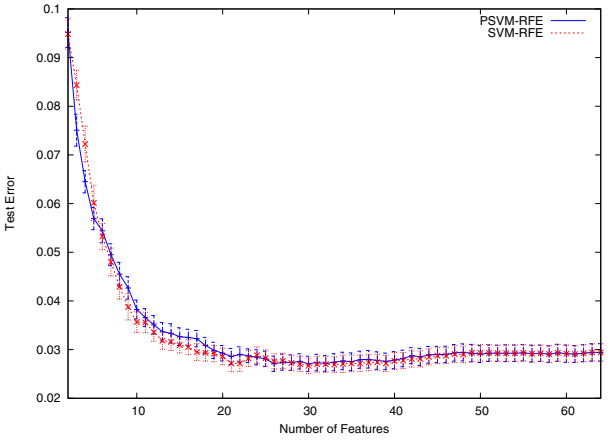


Fig. 4. Same as Figure 2 for the 100 realizations of the Optdigits dataset

wider Qsar dataset we discarded 10 % of the variables each time until 30 variables were left, when we reverted to eliminate the remaining features one by one.

Figures 2, 3 and 4 show the mean test error as a function of the number of variables used for the three datasets. In the first two cases it is evident that the new PSVM-RFE produces lower error solutions when only a few variables are left in the problem, which indicates that our method selects more informative variables in these cases. In the last dataset there is a smaller difference between both methods, also in favor of the new version when modeling with a few variables.

6 Conclusions

In this work we introduced a modification of the basic SVM-RFE method, where the ranking of variables is based on the primal solution of the SVM problem instead of the previous dual-based ranking. Using two artificial datasets we showed that our PSVM-RFE produces a better estimation of the importance of some particular variables when using the Gaussian kernel. In a preliminary study, we showed some examples of real-world datasets in which our method outperforms the traditional, dual-based RFE-SVM.

The use of the primal solution seems more natural, as it considers directly the problem that needs to be solved. As Chapelle remarked in his work, it is possible that the extended use of the dual formulation of SVMs is based in the fact that SVMs were originally introduced in their hard margin formulation, for which the dual optimization seems more natural. SVM-RFE seems to follow this general trend, even if, as we showed, there are no reasons to do so.

Several improvements to SVM-RFE were introduced in the last years. Most of them, as for example Rakotomamonjy's [16] proposal of using the gradient instead of the ranking value, can be easily incorporated to our new formulation. Work in this direction is in progress.

References

1. Alpaydin, E., Kaynak, C.: Cascading classifiers. Kybernetika 34, 369–374 (1998)
2. Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, pp. 144–152. ACM, New York (1992)
3. Caputo, B., Sim, K., Furesjo, F., Smola, A.: Appearance-based object recognition using SVMS: which kernel should i use? In: Proceedings of NIPS Workshop on Statitsical Methods for Computational Experiments in Visual Processing and Computer Vision (2002)
4. Chapelle, O.: Training a support vector machine in the primal. Neural Computation 19(5), 1155–1178 (2007)
5. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
6. Duan, K., Rajapakse, J.C., Wang, H., Azuaje, F.: Multiple SVM-RFE for gene selection in cancer classification with expression data. IEEE Transactions on Nanobioscience 4(3), 228–234 (2005)
7. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
8. Furlanello, C., Serafini, M., Merler, S., Jurman, G.: Entropy-based gene ranking without selection bias for the predictive classification of microarray data. BMC Bioinformatics 4(54) (2003), doi:10.1186/1471–2105–4–54
9. Gentile, C.: Fast feature selection from microarray expression data via multiplicative large margin algorithms. In: Advances in Neural Information Processing Systems, vol. 16 (2004)
10. Granitto, P.M., Gasperi, F., Biasioli, F., Trainotti, E., Furlanello, C.: Modern data mining tools in descriptive sensory analysis: A case study with a random forest approach. Food Quality and Preference 18, 681–689 (2007)
11. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
12. Guyon, I., Weston, S., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46(1-3), 389–422 (2002)
13. Keerthi, S.S., DeCoste, D.: A modified finite newton method for fast solution of large scale linear SVMS. Journal of Machine Learning Research 6, 341–361 (2005)
14. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
15. Mangasarian, O.L.: A finite newton method for classification. Optimization Methods and Software 17, 913–929 (2002)
16. Rakotomamonjy, A.: Variable selection using SVM-based criteria. Journal of Machine Learning Research 3, 1357–1370 (2003)
17. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
18. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
19. Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: Use of the zero norm with linear models and kernel methods. Journal of Machine Learning Research 3 (2003)
20. Xue, Y., Li, Z.R., Yap, C.W., Sun, L.Z., Chen, X., Chen, Y.Z.: Effect of molecular descriptor feature selection in support vector machine classification of pharmacokinetic and toxicological properties of chemical agents. Journal of Chemical Information and Computer Sciences 44, 1630–1638 (2004)
21. Zhou, X., Tuck, D.P.: MSVM-RFE: Extensions of SVM-RFE for multiclass gene selection on DNA microarray data. Bioinformatics 23(9), 1106–1114 (2007)

Pattern Recognition Using Spiking Neurons and Firing Rates

Roberto A. Vázquez

Escuela de Ingeniería - Universidad La Salle
Benjamín Franklin 47 Col. Condesa CP 06140 México, D.F.
ravem@lasallistas.org.mx

Abstract. Different varieties of artificial neural networks have proved their power in several pattern recognition problems, particularly feed-forward neural networks. Nevertheless, these kinds of neural networks require of several neurons and layers in order to success when they are applied to solve non-linear problems. In this paper is shown how a spiking neuron can be applied to solve different linear and non-linear pattern recognition problems. A spiking neuron is stimulated during T ms with an input signal and fires when its membrane potential reaches a specific value generating an action potential (spike) or a train of spikes. Given a set of input patterns belonging to K classes, each input pattern is transformed into an input signal, then the spiking neuron is stimulated during T ms and finally the firing rate is computed. After adjusting the synaptic weights of the neuron model, we expect that input patterns belonging to the same class generate almost the same firing rate and input patterns belonging to different classes generate firing rates different enough to discriminate among the different classes. At last, a comparison between a feed-forward neural network and a spiking neuron is presented when they are applied to solve non-linear and real object recognition problems.

1 Introduction

Artificial neural networks have been applied in a wide range of problems: pattern recognition, forecasting, intelligent control and so on. Perhaps, among the most popular neural network models we could mention the feed-forward neural network trained with the back-propagation algorithm [1,2]. Despite of its power, this type of neural network cannot reach an optimum performance in non-linear problems if it is not well designed. The selection of a learning algorithm and parameters such as number of neurons, number of layers, and the transfer functions, determine the accuracy of the network, resulting in complex architectures to solve the problem [3].

Spiking neuron models have been called the 3rd generation of artificial neural networks [4]. These models increase the level of realism in a neural simulation and incorporate the concept of time. Spiking models have been applied in a wide range of areas from the field of computational neurosciences [5] such as: brain

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 423–432, 2010.

region modeling [6], auditory processing [7,8], visual processing [9,10], robotics [11,12] and so on.

In this paper is shown how a spiking neuron can be applied to solve different linear and non-linear pattern recognition problems. The Izhikevich neuron model [13,14,15] is adopted in this research. A spiking neuron is stimulated during T ms with an input signal and fires when its membrane potential reaches a specific value generating an action potential (spike) or a train of spikes. Given a set of input patterns belonging to K classes, each input pattern is transformed into an input signal, then the spiking neuron is stimulated during T ms and finally the firing rate is computed. After adjusting the synaptic weights of the neuron model, we expect that input patterns belonging to the same class generate almost the same firing rate; on the other hand, we also expect that input patterns belonging to different classes generate firing rates different enough to discriminate among the different classes. At last, a comparison between a feed-forward neural network trained with the well-known backpropagation and Levenberg-Marquartd algorithms and a spiking neuron is presented when they are applied to solve non-linear and real object recognition problems.

2 Izhikevich Neuron Model

A typical spiking neuron can be divided into three functionally distinct parts, called dendrites, soma, and axon. The dendrites play the role of the *input device* that collects signals from other neurons and transmits them to the soma. The soma is the *central processing unit* that performs an important non-linear processing step: if the total input exceeds a certain threshold, then an output signal is generated. The output signal is taken over by the *output device*, the axon, which delivers the signal to other neurons. The neuronal signals consist of short electrical pulses. The pulses, so-called action potentials or spikes, have an amplitude of about 100 mV and typically a duration of 1-2 ms. The form of the pulse does not change as the action potential propagates along the axon. A chain of action potentials emitted by a single neuron is called a spike train (a sequence of stereotyped events which occur at regular or irregular intervals). Since all spikes of a given neuron look alike, the form of the action potential does not carry any information. Rather, it is the number and the timing of spikes which matter. The action potential is the elementary unit of signal transmission. Action potentials in a spike train are usually well separated. Even with very strong input, it is impossible to excite a second spike during or immediately after a first one. The minimal distance between two spikes defines the absolute refractory period of the neuron. The absolute refractory period is followed by a phase of relative refractoriness where it is difficult, but not impossible to excite an action potential [16].

The Izhikevich model

$$\begin{array}{ll} C\dot{v} = k\left(v - v_r\right)\left(v - v_t\right) - u + I & \text{if} v \geq v_{peak} \text{then} \\ \dot{u} = a\left\{b\left(v - v_r\right) - u\right\} & v \leftarrow c, u \leftarrow u + d \end{array} \tag{1}$$

has only nine dimensionless parameters. Depending on the values of a and b, it can be an integrator or a resonator. The parameters c and d do not affect steady-state sub-threshold behavior. Instead, they take into account the action of high-threshold voltage-gated currents activated during the spike, and affect only the after-spike transient behavior. v is the membrane potential, u is the recovery current, C is the membrane capacitance, v_r is the resting membrane potential, and v_t is the instantaneous threshold potential [15].

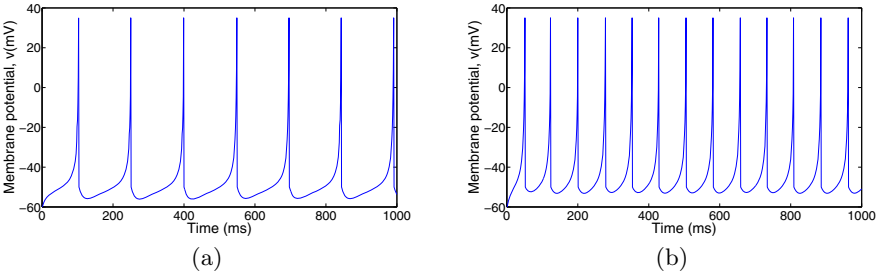
The parameters k and b can be found when one knows the neuron's rheobase and input resistance. The sign of b determines whether u is an amplifying ($b < 0$) or a resonant ($b > 0$) variable. The recovery time constant is a. The spike cutoff value is v_{peak} , and the voltage reset value is c. The parameter d describes the total amount of outward minus inward currents activated during the spike and affecting the after-spike behavior.

Various choices of the parameters result in various intrinsic firing patterns including [13]: RS (*regular spiking*) neurons are the most typical neurons in the cortex, see Fig. 1; IB (*intrinsically bursting*) neurons fire a stereotypical burst of spikes followed by repetitive single spikes; CH (*chattering*) neurons can fire stereotypical bursts of closely spaced spikes; FS (*fast spiking*) neurons can fire periodic trains of action potentials with extremely high frequency practically without any adaptation(slowing down); and LTS (*low-threshold spiking*) neurons can also fire high-frequency trains of action potentials, but with a noticeable spike frequency adaptation.

3 Proposed Method

Before describing the proposed method applied to solve pattern recognition problems, it is important to notice that when the input current signal changes, the response of the Izhikevich neuron also changes, generating different firing rates, see Fig. 1.

The firing rate is computed as the number of spikes generated in an interval of duration T divided by T. The neuron is stimulated during T ms with an input



(a) (b)

Fig. 1. Simulation of the Izhikevich neuron model $100\dot{v} = 0.7\,(v + 60)\,(v + 40) - u + I$, $\dot{u} = 0.03\,\{-2\,(v + 60) - u\}$, if $v \geq 35$ then, $v \leftarrow -50, u \leftarrow u + 100$. (a) Injection of the step of DC current $I = 70$pA. (b) Injection of the step of DC current $I = 100$pA.

signal and fires when its membrane potential reaches a specific value generating an action potential (spike) or a train of spikes.

Let $\left\{\mathbf{x}^i, k\right\}_{i=1}^{p}$ be a set of p input patterns where $k = 1, \ldots, K$ is the class to which $\mathbf{x}^i \in \mathbb{R}^n$ belongs. First of all, each input pattern is transformed into an input signal I, after that the spiking neuron is stimulated using I during T ms and then the firing rate of the neuron is computed. With this information, the average firing rate $\mathbf{AFR} \in \mathbb{R}^K$ and standard deviation $\mathbf{SDFR} \in \mathbb{R}^K$ of each class can be computed.

Finally, we expect that input patterns, which belong to the same class, generate almost the same firing rates (low standard deviation); and input patterns, which belong to different classes, generate firing rates different enough (average spiking rate of each class widely separated) to discriminate among the different classes. In order to achieve that, a training phase is required in the proposed method.

During training phase, the synaptic weights of the model, which are directly connected to the input pattern, are adjusted by means of a differential evolution algorithm.

At last, the class of an input pattern $\tilde{\mathbf{x}}$ is determined by means of the firing rates as

$$cl = \arg \min_{k=1}^{K} \left(|AFR_k - fr| \right) \tag{2}$$

where fr is the firing rate generated by the neuron model stimulated with the input pattern $\tilde{\mathbf{x}}$.

3.1 Generating Input Signals from Input Patterns

Izhikevhic neuron model cannot be directly stimulated with the input pattern $\mathbf{x}^i \in \mathbb{R}^n$, but with an injection current I. Furthermore, to get a firing rate from the neuron model greater than zero, the input signal should be $I > 55$pA.

Since synaptic weights of the model are directly connected to the input pattern $\mathbf{x}^i \in \mathbb{R}^n$, the injection current of this input pattern can be computed as

$$I = \theta + \mathbf{x} \cdot \mathbf{w} \tag{3}$$

where $\mathbf{w}^i \in \mathbb{R}^n$ is the set of synaptic weights of the neuron model and $\theta > 55$ is a threshold which guarantees that the neuron will fire.

3.2 Adjusting Synapses of the Neuron Model

Synapses of the neuron model are adjusted by means of a differential evolution algorithm. Evolutionary algorithms not only have been used to design artificial neural networks [3], but also to evolve structure-function mapping in cognitive neuroscience [17] and compartmental neuron models [18].

Differential evolution is a powerful and efficient technique for optimizing nonlinear and non-differentiable continuous space functions [19]. Differential evolution is regarded as a perfected version of genetic algorithms for rapid numerical

optimization. Differential evolution has a lower tendency to converge to local maxima with respect to the conventional genetic algorithm, because it simulates a simultaneous search in different areas of solution space. Moreover, it evolves populations with smaller number of individuals, and have a lower computation cost.

Differential evolution begins by generating a random population of candidate solutions in the form of numerical vectors. The first of these vectors is selected as the target vector. Next, differential evolution builds a trial vector by executing the following sequence of steps:

1. Randomly select two vectors from the current generation.
2. Use these two vectors to compute a difference vector.
3. Multiply the difference vector by weighting factor F.
4. Form the new trial vector by adding the weighted difference vector to a third vector randomly selected from the current population.

The trial vector replaces the target vector in the next generation if and only if the trial vector represents a better solution, as indicated by its measured cost value computed with a fitness function. Differential evolution repeats this process for each of the remaining vectors in the current generation. Differential evolution then replaces the current generation with the next generation, and continues the evolutionary process over many generations.

In order to achieve that input patterns, which belong to the same class, generate almost the same firing rates (low standard deviation), and input patterns, which belong to different classes, generate firing rates enough different (average spiking rate of each class widely separated) to discriminate among the different classes, the next fitness function was proposed

$$f = \frac{1}{dist\,(\mathbf{AFR})} + \sum_{k=1}^{K} SDFR_k \tag{4}$$

where $dist\,(\mathbf{AFR})$ is the summation of the Euclidian distances among the average firing rate of each class.

4 Experimental Results

To evaluate and compare the accuracy of the proposed method, against feedforward neural networks, several experiments were performed using 3 datasets. Two of them were taken from UCI machine learning benchmark repository [20]: iris plant and wine datasets. The other one was generated from a real object recognition problem.

The iris plant dataset is composed of 3 classes and each input pattern is composed of 4 features. The wine dataset is composed of 3 classes and each input pattern is composed of 13 features.

For the case of the real object recognition problem, a dataset was generated from a set of 100 images which contains 5 different objects whose images are shown in Fig. 2 [21]. Objects were not recognized directly from their images,

(a) (b) (c) (d) (e) (f)

Fig. 2. (a)-(c) Some of the images used to train the proposed method. (d)-(f) Some of the images used to test the proposed method.

but by an invariant description of each object. Several images of each object in different positions, rotations and scale changes were used. To each image of each object a standard thresholder [22] was applied to get its binary version. Small spurious regions were eliminated from each image by means of a size filter [23]. Finally, the 7 well-known Hu geometric invariants, to translations, rotations and scale changes [24], were computed to build the object recognition dataset.

All data sets were partitioned by means of a 5-fold-cross validation strategy.

The parameters for the Izhikevich neuron were defined as $C = 100$, $v_r = -60$, $v_t = -40$, $v_{peak} = 35$, $k = 0.7$, $a = 0.03$, $b = -2$, $c = -50$, and $d = 100$. The Euler method was used to solve the differential equation of the model with $dt = 1$. The parameter to compute input current I from the input pattern was set to $\theta = 100$ with a duration of $T = 1000$. For the case of the differential evolution algorithm, $NP = 40$, $MAXGEN = 1000$, $F = 0.9$, $XMAX = 10$, $XMIN = -10$ and $CR = 0.5$.

The classic back-propagation and Levenberg-Marquardt algorithms were used to train the feed-forward neural network. The number of generations was set to 10000 and learning rate $\alpha = 0.01$. Concerning to the architecture of the feed-forward neural network, one hidden layer composed of 13 *hyperbolic tangent* neuron and an output layer composed of *linear* neurons were used in all experiments.

The stop criterion for the three algorithms was the number of generations or the minimum error which was set to $e = 0$.

The accuracy (classification rate), achieved with the proposed method, was computed as the number of input patterns correctly classified divided by the total number of tested input patterns. To validate the accuracy of the proposed method 20 experiments over each dataset were performed. The same metric and number of experiments were used to measure the accuracy of the feed-forward neural network trained with the two different algorithms. Something important to notice is that in each experiment a new set of partitions over each dataset was generated by means of the 5-fold-cross validation strategy.

The experimental results, obtained with the iris plant dataset, are shown in Fig 3. The reader can appreciate in Table 1 that the Izhikevich neuron trained with the proposed method provided better results than the feed-forward neural network trained with the two classical algorithms. Furthermore, standard deviation from all experiments was really small compared against that obtained using feed-forward neural networks. Something that should be remarked is that while the feed-forward neural network was composed of 16 neurons, the proposed method only used one Izhikevich neuron.

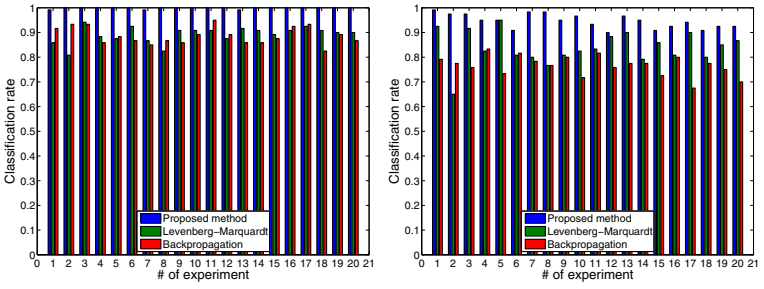


Fig. 3. Experimental results obtained with the iris plant dataset. (a) Accuracy of the proposed method during training phase. (b) Accuracy of the proposed method during testing phase.

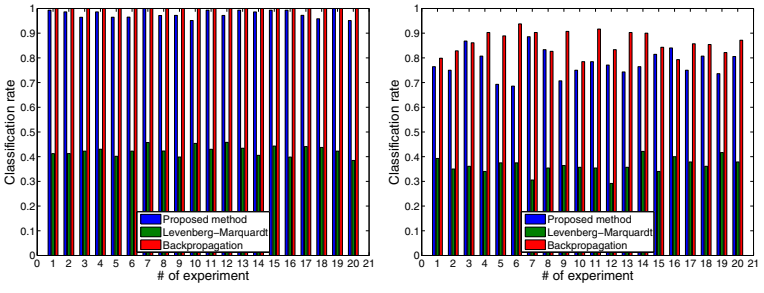


Fig. 4. Experimental results obtained with the wine dataset. (a) Accuracy of the proposed method during training phase. (b) Accuracy of the proposed method during testing phase.

The experimental results, obtained with the wine dataset, are shown in Fig 4. The reader can appreciate in Table 1 that the Izhikevich neuron trained with the proposed method also provided better results than the feed-forward neural network trained with the backpropagation algorithm; but Levenberg-Marquartd method provided better results with a neural network composed of 16 neurons. The average classification rate obtained during testing phase was not as good as that obtained during training phase.

The experimental results, obtained with the object recognition dataset, are shown in Fig 5. The reader can appreciate in Table 1 that results provided by the Izhikevich neuron are better compared to those obtained with the feed-forward neural network. Although the feed-forward neural network was composed with 18 neurons and trained during 10000 iterations, the proposed method provided better results only with 1000 iterations.

In Table 1 and 2 the average and standard deviation classification rate computed from all experimental results are shown. The results obtained with the

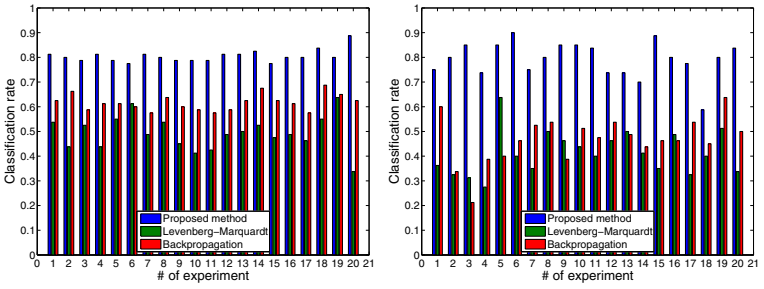


Fig. 5. Experimental results obtained with the object recognition dataset. (a) Accuracy of the proposed method during training phase. (b) Accuracy of the proposed method during testing phase.

Table 1. Average accuracy provided by the methods using different databases

Dataset	Back-propagation		Levenberg-Marquartd		Proposed method	
	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.	Tr. cr.	Te. cr.
Iris plant	0.8921	0.8383	0.8867	0.7663	0.9988	0.9458
Wine	0.4244	0.3637	1	0.8616	0.9783	0.7780
Object rec.	0.4938	0.4125	0.6169	0.4675	0.8050	0.7919

Tr. cr = Training classification rate, Te. cr. = Testing classification rate.

Table 2. Standard deviation obtained through the different experiments

Dataset	Back-propagation		Levenberg-Marquartd		Proposed method	
	Tr.	Te.	Tr.	Te.	Tr.	Te.
Iris plant	0.0684	0.1015	0.0676	0.0918	0.0064	0.0507
Wine	0.0456	0.0674	0	0.0750	0.0359	0.0930
Object rec.	0.1586	0.1689	0.0907	0.1439	0.0770	0.0793

spiking neuron model, trained with the proposed method, improve the results obtained with feed-forward neural networks, when are applied to non-linear problems. Furthermore, the standard deviations, computed from all experiments, support the stability of the proposed method. While feed-forward neural networks provided different results through each experiment, the proposed method provided similar results.

These preliminary results suggest that spiking neurons can be considered as an alternative way to perform different pattern recognition tasks.

We can also conjecture that if only one neuron is capable to solve pattern recognition problems, perhaps several spiking neurons working together can improve the experimental results obtained in this research. However, that is something that should be proved.

5 Conclusions

In this paper a new method to apply a spiking neuron in a pattern recognition task was proposed. This method is based on the firing rates produced with an Izhikevich neuron when is stimulated. The firing rate is computed as the number of spikes generated in an interval of duration T divided by T. A spiking neuron is stimulated during T ms with an input signal and fires when its membrane potential reaches a specific value generating an action potential (spike) or a train of spikes.

The training phase of the neuron model was done by means of a differential evolution algorithm. After training, we observed that input patterns, which belong to the same class, generate almost the same firing rates (low standard deviation) and input patterns, which belong to different classes, generate firing rates different enough (average spiking rate of each class widely separated) to discriminate among the different classes.

Through several experiments we observed that the proposed method, significantly improve the results obtained with feed-forward neural networks, when are applied to non-linear problems. On the other hand, the standard deviations computed from all experiments, support the stability of the proposed method. While feed-forward neural networks provided different results through each experiment, the proposed method provided similar results. Finally, we can conclude that spiking neurons can be considered as an alternative tool to solve pattern recognition problems.

Nowadays, we are analyzing the different behaviors of the Izhikevich model in order to determine which configuration is the most suitable for a pattern recognition task. In addition, we are testing different fitness functions to improve the results obtained in this research. Furthermore, we are researching different alternatives of combining several Izhikevich neurons in one network to improve the results obtained in this research and then apply it in more complex pattern recognition problems such as face, voice and 3D object recognition.

Acknowledgements

The authors thank Universidad La Salle for the economical support under grant number ULSA I-113/10.

References

1. Anderson, J.A.: Introduction to Neural Networks. MIT Press, Cambridge (1995)
2. Werbos, P.J.: Backpropagation through time: What it does and how to do it. Proc. IEEE 78, 1550–1560 (1990)
3. Garro, B.A., Sossa, H., Vazquez, R.A.: Design of Artificial Neural Networks using a Modified Particle Swarm Optimization Algorithm. IJCNN, 938–945 (2009)
4. Maass, W.: Networks of spiking neurons: the third generation of neural network models. Neural Networks 10(9), 1659–1671 (1997)

5. Rieke, F., et al.: Spikes: Exploring the Neural Code. Bradford Book (1997)
6. Hasselmo, M.E., Bodelon, C., et al.: A Proposed Function for Hippo-campal Theta Rhythm: Separate Phases of Encoding and Retrieval Enhance Re-versal of Prior Learning. Neural Computation 14, 793–817 (2002)
7. Hopfield, J.J., Brody, C.D.: What is a moment? Cortical sensory integration over a brief interval. PNAS 97(25), 13919–13924 (2000)
8. Loiselle, S., Rouat, J., Pressnitzer, D., Thorpe, S.: Exploration of rank order coding with spiking neural networks for speech recognition. IJCNN 4, 2076–2080 (2005)
9. Azhar, H., Iftekharuddin, K., et al.: A chaos synchronization-based dynamic vision model for image segmentation. IJCNN 5, 3075–3080 (2005)
10. Thorpe, S.J., Guyonneau, R., et al.: SpikeNet: Real-time visual processing with one spike per neuron. Neurocomputing 58(60), 857–864 (2004)
11. Di Paolo, E.A.: Spike-timing dependent plasticity for evolved robots. Adaptive Behavior 10(3), 243–263 (2002)
12. Floreano, D., Zufferey, J., et al.: From wheels to wings with evolutionary spiking neurons. Artificial Life 11(1-2), 121–138 (2005)
13. Izhikevich, E.M.: Simple model of spiking neurons. IEEE Trans. on Neural Networks 14(6), 1569–1572 (2003)
14. Izhikevich, E.M.: Which model to use for cortical spiking neurons? IEEE Trans. on Neural Networks 15(5), 1063–1070 (2004)
15. Izhikevich, E.M.: Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. The MIT Press, Cambridge (2007)
16. Gerstner, W., et al.: Spiking Neuron Models. Cambridge University Press, Cambridge (2002)
17. Frias-Martinez, E., Gobet, F.: Automatic generation of cognitive theories using genetic programming. Minds and Machines 17(3), 287–309 (2007)
18. Hendrickson, E., et al.: Converting a globus pallidus neuron model from 585 to 6 compartments using an evolutionary algorithm. BMC Neurosci. 8(s2), P122 (2007)
19. Price, K., Storn, R.M., Lampinen, J.A.: Diffentential evolution: a practical approach to global optimization. Springer, Heidelberg (2005)
20. Murphy, P.M., Aha, D.W.: UCI repository of machine learning databases. Dept. Inf. Comput. Sci., Univ. California, Irvine, CA (1994)
21. Vazquez, R.A., Sossa, H.: A new associative model with dynamical synapses. Neural Processing Letters 28(3), 189–207 (2008)
22. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. on SMC 9(1), 62–66 (1979)
23. Jain, R., et al.: Machine Vision. McGraw-Hill, New York (1995)
24. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Trans. on Information Theory 8, 179–187 (1962)

Interaction Graphs for Multivariate Binary Data

Johan Van Horebeek[1] and Jesús Emeterio Navarro-Barrientos[2]

[1] Centro de Investigación en Matemáticas, CIMAT,
Ap. Postal 402, 36000 Guanajuato, GTO, Mexico
[2] School of Mathematical and Statistical Sciences, Arizona State University,
P.O. Box 871804, Tempe, AZ
horebeek@cimat.mx, jnavarrob@asu.edu

Abstract. We define a class of graphs that summarize in a compact visual way the interaction structure between binary multivariate characteristics. This allows studying the conditional dependency structure between the underlying stochastic variables at a finer scale than with classical probabilistic Graphical Models. A model selection strategy is derived based on an iterative optimization procedure and the consistency problem is discussed. We include the analysis of two data-sets to illustrate the proposed approach.

Keywords: Graphical Models, conditional dependency, data visualization.

1 Introduction

Graphical elements are of increasing importance in a variety of methods to study the interaction structure of multivariate data. In this context, graphs as employed in *Graphical Models* [9], have turned out to be useful instruments. Especially in Machine Learning they have become a standard tool in areas like probabilistic expert systems, statistical learning or data mining [3].

Graphical Models were originally conceived to describe families of multivariate distributions by means of a graph where nodes represent the variables and the absence of edges between variables reflects particular conditional independencies in the underlying interaction structure.

The use of graphs contributed significantly to the success of these models: on one hand, they provide an intuitively clear interface between the data, its underlying distribution and the user; on the other hand they allow to formulate and derive many efficient inference algorithms in terms of characteristics of the graph like cliques and separability.

Despite of the above, relatively little attention has been paid to how to make Graphical Models more informative and suitable for exploratory purposes. Probably the oldest example can be found in [14] where the width of the edges is made variable in order to reflect the support of the corresponding conditional independence hypothesis. This serves as a guide in data understanding and model building. However, it reflects only an overall summary at the level of variables.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 433–442, 2010.

Split Models [7] and *Generalized Graphical Models* [12] extend Graphical Models by introducing a way to incorporate information about conditional (in)dependencies which depend on the values of the variables in the conditional part. The latter adds labels to the edges, the former builds a tree of graphs; an example is shown in Fig. 1 for the case $X_1 \perp X_2 | X_3 = 0$ and $X_1 \not\perp X_2 | X_3 = 1$, i.e., only when X_3 equals 0, the variables X_1 and X_2 are conditionally independent. A similar approach is taken in [7]; it is directly formulated in terms of the underlying clique structure but with no visual representation.

In other approaches like [8] additional information is included about the presence of symmetry in the underlying interaction structure.

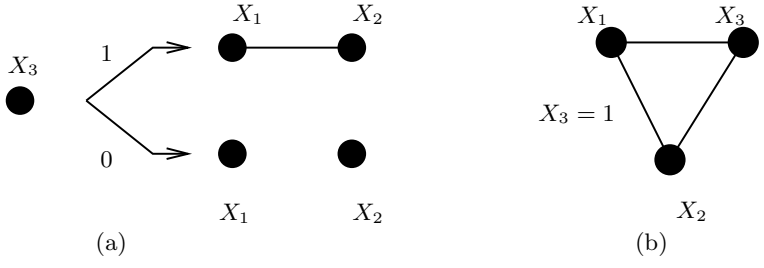


Fig. 1. Visual representations for the case $X_1 \perp X_2 | X_3 = 0$ and $X_1 \not\perp X_2 | X_3 = 1$.: (a) Split Model and (b) Generalized Graphical Model.

In this contribution we extend [10] and [12], following the original motivation for Graphical Models and using ideas proposed by e.g. [6]; this will lead to a class of graphs called *interaction graphs*. Our approach characterizes the interaction structures between multivariate variables at a finer scale including more information in the graph to facilitate model selection.

We consider the particular case of the multivariate binary stochastic variables because of its importance in many Computer Science applications.

In the sequel X denotes a d-dimensional binary stochastic variable $(X_1, \cdots, X_d)$ where $X_i \in \{0, 1\}$ and we suppose that all cell probabilities $p_{x_1,\cdots,x_d} := P(X_1 = x_1, \cdots, X_d = x_d)$ are strictly positive. The vector with all cell probabilities will be denoted by P. The symbol '-' before an index of a vector refers to the exclusion of the corresponding entry: e.g. $X_{-i,-j} = \{X_k | k = 1, ..., d; k \neq i, j\}$. To refer simultaneously to a set A of entries of a vector we will write X_A. Finally, if the conditional independence $X_i \perp X_j | X_{-i,-j} = x_{-i,-j}$ is valid for all values $x_{-i,-j}$, we write $X_i \perp X_j | X_{-i,-j}$.

The paper is organized as follows: Section 2 reviews briefly some aspects of dependencies between variables and of Graphical Models. Section 3 introduce Interaction Graphs and Section 4 shows two applications. Finally, Section 5 summarizes our main conclusions and discusses areas of further study.

2 Dependencies and Graphical Models

2.1 Odds-ratio

The *(log) odds-ratio* is a fundamental concept to quantify the dependency between binary variables and is the starting point of other important association measures like, e.g., Yule's Q and Y measure (see [2]). It is defined as:

$$\theta_{i,j}(x_{-i,-j}) = \frac{P(X_i=1,X_j=1,X_{-i,-j}=x_{-i,-j})P(X_i=0,X_j=0,X_{-i,-j}=x_{-i,-j})}{P(X_i=1,X_j=0,X_{-i,-j}=x_{-i,-j})P(X_i=0,X_j=1,X_{-i,-j}=x_{-i,-j})}.$$

In the binary case, the odds-ratio $\theta_{i,j}(x_{-i,-j})$ summarizes completely the conditional interaction between two variables:

$$X_i \perp X_j | X_{-i,-j} = x_{-i,-j} \text{ iff } \theta_{i,j}(x_{-i,-j}) = 1 \text{ iff } \log(\theta_{i,j}(x_{-i,-j})) = 0 \quad (1)$$

As will be shown in Section 3, for our purpose it is useful and illustrative to represent a discrete binary distribution as a hypercube where the vertices correspond to the cell probabilities. As shown in Fig. 2, each odds-ratio is in 1-1 correspondence with a particular face of the hypercube, defined by the cell probabilities associated with that face.

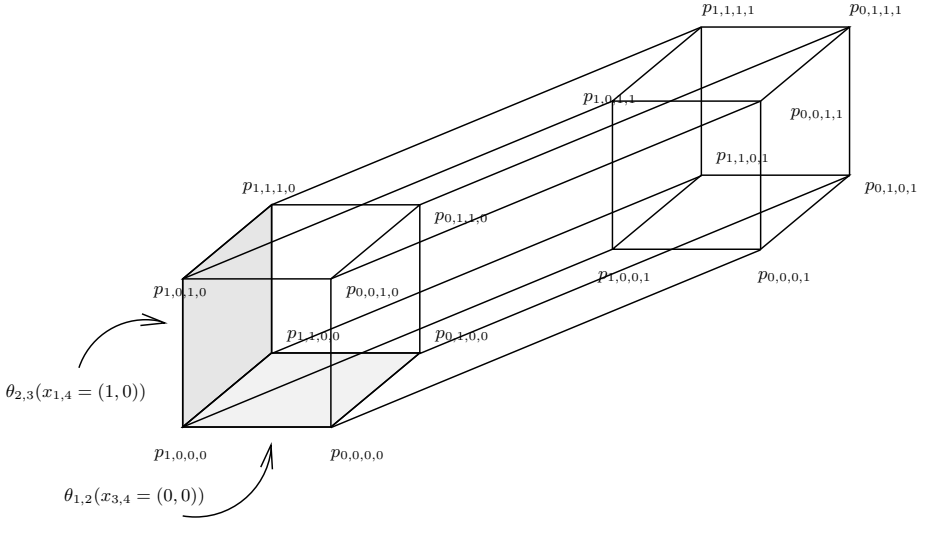


Fig. 2. Representation of a multivariate binary distribution ($d = 4$)

In the sequel, we will also make use of the fact that $\log(\theta_{i,j}(x_{-i,-j})) = 0$ imposes an orthogonality constraint of the vector $\log P$ to a vector v; e.g., if $d = 3$:

$$\log(\theta_{1,2}(x_3 = 0)) = 0 \;\Leftrightarrow\; X_1 \perp X_2 | X_3 = 0 \;\Leftrightarrow \tag{2}$$
$$(\log p_{1,1,0}, \log p_{0,0,0}, \log p_{1,0,0}, \log p_{0,1,0}, \log p_{1,1,1}, \log p_{0,0,1}, \log p_{1,0,1}, \log p_{0,1,1})$$
$$\perp (1, 1, -1, -1, 0, 0, 0, 0)$$

As the odds-ratios are related to each other, they do not constitute an independent set of parameters and we will need other parameterizations of P; probably the best known is the *Log-linear model* which we explain briefly in the next section.

2.2 Graphical Models

Graphical Models are a subclass of Log-linear models. The latter were introduced by Birch [1] as of particular way of parameterizing $P(X_1 = x_1, \cdots, X_d = x_d)$. The most generic form for the binary case is

$$\log P(X_1 = x_1, \cdots, X_d = x_d) = \sum_i \beta_i f(x_{A_i}) \;\text{ with } A_i \subset \{1, \cdots, d\} \tag{3}$$

or, equivalently written in vector form, for a fixed *design matrix* $\mathbb{A}$:

$$\log P = \mathbb{A} \cdot \beta. \tag{4}$$

Graphical Models are Log-linear models satisfying a set of independence statements S of the form

$$X_i \perp X_j | X_{-i,-j}, \tag{5}$$

i.e., independencies which do not depend on the values of the variables in the conditional part.

Given S, a graph is constructed: each X_i represents a node in the graph and the absence of an edge between node i and j corresponds to an independence of the form specified in Eq. (5). One tacitly assumes that all independencies of the form (5) that do not belong to S are not present in the distribution (4). See Fig. 6 for an example. The graph is used as a summary of the main structure of the data.

One of the advantages of working only with independencies of the form (5) is that they traduce immediately in straightforward restrictions on which terms should enter in (3). A disadvantage is that this limitation often will be too restrictive, as shown in the following example.

To this end, consider the following data-set from a survey about the circumstances in which accidents occurred between American football players [4]. Table 1 shows the data-set for the following three variables: X_1 the accident occurred in a defensive play (value 0) or in an attack (value 1), X_2 the accident occurred while throwing the ball (value 0) or not (value 1) and X_3 the accident occurred in a tackle (value 0) or in a block (value 1). It can be shown that for this data-set the only Graphical Model that can be accepted corresponds to a complete connected graph (all hypothesis tests of the form (5) have

Table 1. Contingency table of accidents occurred to football players

	$X_3 = 0$		$X_3 = 1$	
	$X_2 = 0$	$X_2 = 1$	$X_2 = 0$	$X_2 = 1$
$X_1 = 0$	125	85	129	31
$X_1 = 1$	216	62	61	16

a p-value ≤ 0.01). Nevertheless, if we look at the estimated log odds-ratios and their 95% confidence intervals:

$\log(\theta_{1,2}(x_3 = 0)) = -0.86$ $[-1.26, -0.46]$ $\log(\theta_{1,2}(x_3 = 1))$= $\mathbf{0.09}$ $[\mathbf{-0.59}, \mathbf{0.76}]$
$\log(\theta_{1,3}(x_2 = 0)) = -1.3$ $[-1.67, -0.92]$ $\log(\theta_{1,3}(x_2 = 1))$= $\mathbf{-0.34}$ $[\mathbf{-1.03}, \mathbf{0.34}]$
$\log(\theta_{2,3}(x_1 = 0)) = -1.04$ $[-1.52, -0.56]$ $\log(\theta_{2,3}(x_1 = 1)) = \mathbf{-0.09}$ $[\mathbf{-0.71}, \mathbf{0.53}]$

we observe that some log odds-ratios (marked in bold) might be zero; hence some conditional independencies for the particular values of some variables might be plausible. As they cannot be captured by Graphical Models, the question raises how this might be included in those graphs. This will be explained in the next section.

3 Interaction Graphs

3.1 Visual Representation

The underlying idea of Interaction Graphs is to incorporate in a graph information about the (log) odds-ratios to reflect the dependency structure.

We start with a graph as used in a classical Graphical Model and subdivide the edge between each pair of nodes i and j in $l = 2^{d-2}$ segments, each one associated with a particular assignation of values to $X_{-i,-j}$. Fig. 3 shows an example of an edge segmentation in a data-set with five variables where the correspondence is made by means of a lexicographic ordering of the values of $x_{-1,-2}$.

An absence of a segment corresponds to an independence of the form as in Eq. (1). As illustrated in Fig. 4, the lexicographic ordering has the advantage that independencies of the form

$$X_1 \perp X_2 | X_{-1,-2,A} = x_{-1,-2,-A}, X_A, \tag{6}$$

correspond to easily detectable patterns as those shown in Fig. 4.

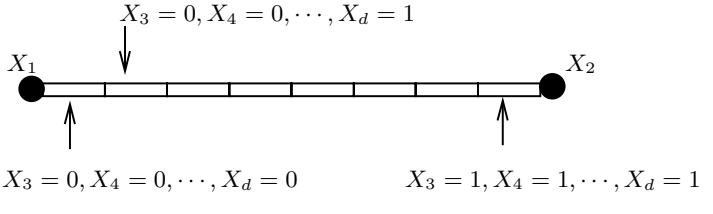


Fig. 3. Edge segmentation between variables for $X_1 \perp X_2 | X_{-1,-2} = x_{-1,-2}$ when $d = 5$

Fig. 4. Edge segmentation patterns in a data-set with five variables for the following conditional independence between X_1 and X_2, respectively: $X_1 \perp X_2 | X_5 = 0, X_{3,4}$; $X_1 \perp X_2 | X_4 = 1, X_{3,5}$ and $X_1 \perp X_2 | X_3 = 0, X_{4,5}$.

The graph will not only be used to summarize a given set of independencies but also to suggest new ones. To this end, we attach to the segments a color and make their width variable: the width is inversely proportional to the p-value of the test of the corresponding conditional independence using a chosen goodness-of-fit statistic. The segment is colored light gray if the corresponding p-value is above a user specified threshold, meaning that the segment may be taken out of the graph; otherwise, it is colored green or red depending whether the odds-ratio is significantly greater than or smaller than one, meaning a positive or negative correlation, respectively. See Fig. 5 for an example.

3.2 Estimation in Interaction Graphs

In order to adjust a particular model defined by a given set of independencies as in Eq. (1), we make use of the fact that each conditional independence Eq. (5) can be expressed as an orthogonality constraint of the vector $\log P$ to a vector v_i as explained in Section 2.1. In this way, we obtain a set $V = \{v_1, \cdots, v_m\}$ to which $\log P$ should be orthogonal. Next, we calculate a basis for the space $V^{\perp}$. This is used to define the matrix $\mathbb{A}$ in Eq. (4).

Using a Newton-Raphson type optimization algorithm we calculate the maximum likelihood estimator for β (taking also into account the normalization constraint on P). This value is used in a Power Divergence Goodness of Fit statistic [11] to calculate the p-value of the hypothesis that the given set of independencies hold.

3.3 Consistency Problem

Opposite to Graphical Models, not every graph corresponds to a valid model. This is a consequence of the odds-ratios being related to each other. In the multiplication of two odds-ratios whose corresponding faces in the hypercube (cf. Fig. 2) have an edge in common, the middle terms will cancel out, e.g.,

$$\theta_{1,2}(x_{3,4} = (0,0)).\theta_{2,3}(x_{1,4} = (1,0)) = \frac{p_{0,0,0,0}p_{1,1,0,0}}{p_{0,1,0,0}p_{1,0,0,0}} \frac{p_{1,0,0,0}p_{1,1,1,0}}{p_{1,1,0,0}p_{1,0,1,0}} = \frac{p_{0,0,0,0}p_{1,1,1,0}}{p_{0,1,0,0}p_{1,0,1,0}}.$$

As a consequence, if we repeat this for a whole sequence, eventually an odds-ratio can be expressed in terms of others. E.g., it is easy to check that:

$$\theta_{1,2}(x_3 = 0) = (\theta_{1,3}(x_2 = 1))^{-1}\, \theta_{1,2}(x_3 = 1)\, \theta_{1,3}(x_2 = 0).$$

This means that if the odds-ratios on the RHS are equal to 1, the LHS also equals one. In other words, independencies (associated to the RHS) will imply another one (associated to the LHS). Using a representation like in Fig. 2, the RHS corresponds to a loop of faces with the associated odds-ratios equal to 1, starting and ending on some face (whose odds-ratio corresponds to the LHS).

These relationships between the odds-ratios lead to a way to detect implied independencies (and hence inconsistencies) for a given set of independencies.

4 Examples of Using Interaction Graphs

Fig. 5 (a) shows an Interaction Graph for the saturate model of the data-set *Football Accidents*. The red segments indicate negative (conditional) correlations between the variables. Moreover, there are three gray segments suggesting that the following independencies are plausible:

$$X_1 \perp X_2 | X_3 = 1 \quad X_1 \perp X_3 | X_2 = 1 \quad X_2 \perp X_3 | X_1 = 1.$$

Fig. 5 (b) shows the final Interaction Graph.

The second example we consider is the data-set *Women and Mathematics* [5]. The data-set is based on a survey of 1190 New Jersey high school students about their interest in Mathematics; some of the students attended a conference that promotes Mathematics. The variables are: X_1 type of school (0=suburban, 1=urban), X_2 sex of the person (0=female, 1=male), X_3 assistance to the conference (0= yes, 1=no), X_4 Future Plans (0=University, 1=work), X_5 course

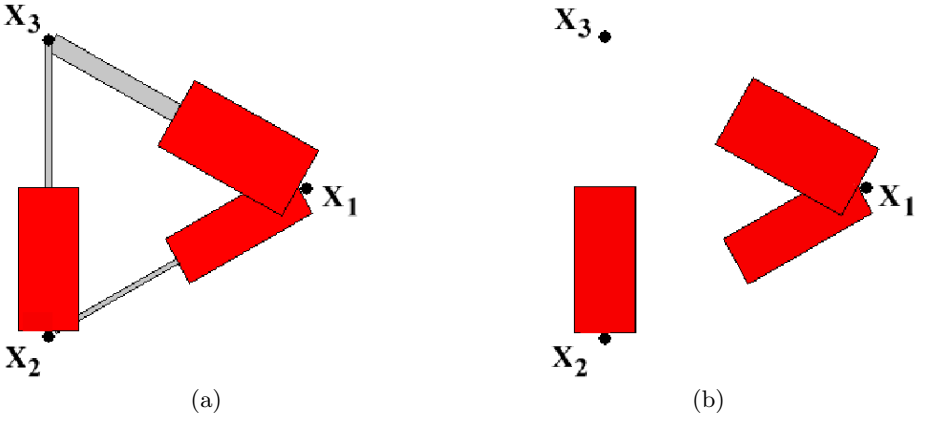


Fig. 5. Interaction Graph for the data-set *Football Accidents*: (a) saturated model and (b) final accepted model with a p-value of 0.28946

preferences (0=Mathematics, 1=Liberal Arts), X_6 necessity of Mathematics in the future (0=agree, 1=disagree).

Fig. 6 is a plausible classical Graphical Model (it has a p-value of 0.18). The graph shows e.g. that X_3 "assistance to the conference" is independent to all other variables. On the other hand, there seems to be a statistical significant dependence between e.g. X_1 "type of school" and X_4 "future plans", and between X_6 "necessity of Mathematics in the future" and X_4 "future plans". Fig. 7 shows an Interaction Graphs for these data; its p-value is 0.1228.

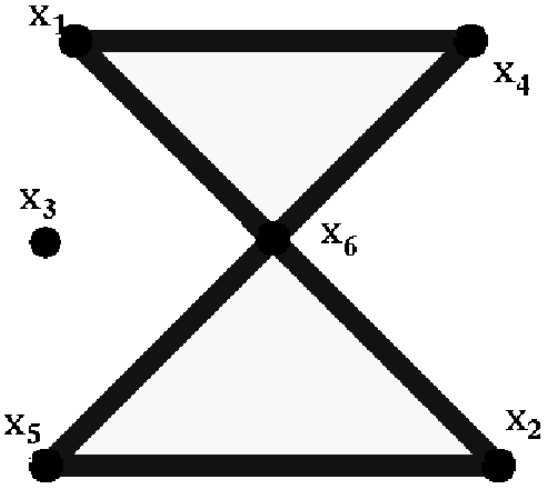


Fig. 6. Classical Graphical Model for data survey *Women and Mathematics*

Apart of the independencies from the previous graph, we learn in addition that the correlation between X_1 and X_4 is mainly negative. The dependency between X_1 and X_4 seems to depend on the value of a particular variable:

$$X_4 \not\perp X_6 | X_1 = 1, X_{2,3,5} \quad X_4 \perp X_6 | X_1 = 0, X_{2,3,5}.$$

Keeping in mind the patterns of Fig. 4, one discovers:

$$X_1 \perp X_6 | X_4 = 0, X_{2,3,5}$$
$$X_2 \perp X_6 | X_5 = 1, X_{1,3,4}$$
$$X_4 \perp X_6 | X_1 = 0, X_{2,3,5}$$

In practice, the software guides the user in identifying the corresponding variables. Observe that by weakening the independencies Graphical Models are working with, the set of potential distributions increases dramatically; this leads to a practical limit for the visualization.

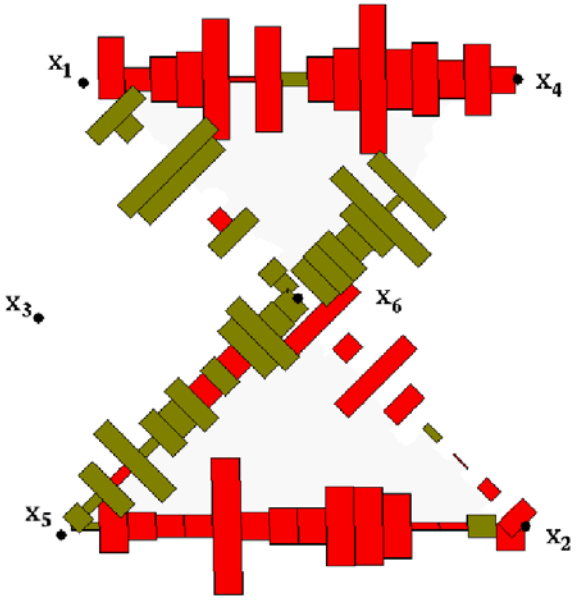


Fig. 7. Interaction Graph for data-set *Women and Mathematics* (its p-value is 0.1228)

5 Conclusions

In close connection with the basic philosophy of Graphical Models, we proposed in this paper a particular class of graphs that facilitate the study of conditional interaction structures between binary multivariate stochastic variables. To this end, we proposed the use of odds-ratios as the basic features of the underlying distribution. In the first place, the resulting graph pretends to offer an informative tool for exploratory purposes. A next step is to develop appropriate methods for probabilistic inference as required by e.g. probabilistic expert systems.

References

1. Birch, M.W.: Maximum Likelihood in Three-way Contingency Tables. Journal of the Royal Statistical Society B 25, 220–233 (1963)
2. Bishop, Y., Fienberg, S., Holland, W.: Discrete Multivariate Analysis. MIT Press, Cambridge (1990)
3. Borgelt, C., Kruse, R.: Graphical Models: Methods for Data Analysis and Mining. John Wiley, Chichester (2002)
4. Buckley, W.: Concussions in Football: a Multivariate Analysis. American Journal of Sport Medicine 16, 609–617 (1988)
5. Fowlkes, E.B., Freeny, A.E., Landwehr, J.M.: Evaluating Logistic Models for Large Contingency Tables. JASA 83, 611–622 (1989)
6. Friendly, M.: Visualizing Categorical Data. SAS Institute, Cary (2000)

7. Højsgaard, S.: Split Models for Contingency Tables. Computational Statistics and Data Analysis 42, 621–645 (2003)
8. Højsgaard, S.: Inference in Graphical Gaussian Models with Edge and Vertex Symmetries with the gRc Package for R. Journal of Statistical Software 23 (2007)
9. Koller, D., Friedman, N.: Probabilistic Graphical Models. MIT Press, Cambridge (2009)
10. Navarro-Barrientos, J.E.: Modelos Gráficos Clásicos y Generalizados para el Análisis de Datos Binarios y su Aplicación en Datos de Accesos de Internet. M.Sc. Thesis CIMAT (2001)
11. Read, T., Cressie, N.: Goodness of Fit Statistics for Discrete Multivariate Data. Springer, Heidelberg (1988)
12. Teugels, J.L., Van Horebeek, J.: Generalized Graphical Models for Discrete Data. Statistics & Probability Letters 38, 41–47 (1998)
13. Wermuth, N., Lauritzen, S.: Graphical and Recursive Models for Contingency Tables. Biometrika 70, 537–552 (1983)
14. Whittaker, J.: Graphical Models in Applied Multivariate Statistics. Wiley Series in Probability and Mathematical Statistics (1989)

Detecting Drifts in Multi-Issue Negotiations

Gislaine M. Nalepa, Bráulio C. Ávila,
Fabrício Enembreck, and Edson E. Scalabrin

PUCPR, Pontifical Catholic University of Paraná
PPGIA, Graduate Program on Applied Computer Science
R. Imaculada Conceição, 1155 Curitiba PR Brazil
{gislaine,avila,fabricio,scalabrin}@ppgia.pucpr.br
http://www.ppgia.pucpr.br

Abstract. In an automated negotiation process, it is usual for one party not to reveal their preferences to its opponent. In an attempt to adjust itself to this process, each *trade agent* can be endowed with capabilities of learning and detecting its opponent's changing preferences. This paper presents techniques for *drift detection* that are useful in this scenario: the Instance-Based Learning algorithms (IB1, IB2 and IB3) and the Bayesian Networks. Theoretically, a group of expert agents is able to achieve better results than an individual, therefore, the DWM is part of these studies, once it provides an effective strategy for integrating the decision of several *expert agents*. The experiments performed revolve around the settings DWM-IB3 and DWM-Bayesian Network. The performance of each system was evaluated for gradual, moderate and abrubt drifts in concept, and the results showed that both settings are able to efficiently detect drifts in the preferences of the opponent.

1 Introduction

Software agents are computer programs that can replace humans in complex activities that require knowledge, experience and reasoning, such as having skills to solve conflicts through negotiation processes. A negotiation can be defined as a form of decision making involving two or more agents, which cannot make decisions independently; such agents are required to make concessions to reach an agreement [8]. Several studies [2, 4, 5, 9] were performed to improve an agent's negotiation capability aiming high performance. Those studies invest in learning algorithms that enable agents to learn the preferences of their opponents and to detect when these preferences have changed, adapting themselves automatically to them. *Drift detection* technique has proven itself useful in this scenario. It aims to quickly detect the concept drift so that fewer mistakes are committed, and so actual drifts are distinguished from noise.

This paper aims to present some techniques for *drift detection*. Therefore, an algorithm belonging to the Instance Based Learning (IBL) family [1], a Bayesian learning technique and an algorithm capable of controlling a group of learners are presented. Within this context, two systems were developed, in which the DWM

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 443–452, 2010.

(Dynamic Weighted Majority) was used along with a learning algorithm, namely: IB3 and Bayesian Network. The objective here is to compare the efficiency of the DWM-IB3 setting of Enembreck [5] with the efficiency of the DWM-BN (Bayesian Network) regarding the concept drift.

Section 2 presents the IBL family, detailing the operation of IB3. Section 3 introduces the concepts of Bayesian Networks, detailing types of construction, inference and learning. The DWM algorithm is presented in Section 4. The developed systems are in Section 5 and the experiments' results are discussed in Section 6. Finally, conclusions are presented.

2 Instance-Based Learning Algorithms

IBL is a methodology in which the predictions are performed using specific instances. The algorithms that are part of this family are: IB1, IB2, IB3, IB4 and IB5, however, the latter two are not detailed in this paper. They use the technique of "Nearest Neighbour" (NN) to classify each new instance. The set of training examples of an IBL algorithm expresses the concept description; such a description is determined by means of similarity functions and classification, defined by Aha [1].

2.1 The IB1 Algorithm

Figure 1 ilustrates the operation of IB1, the simplest algorithm of the IBL family. In the first activity, a similarity function is used to calculate the similarity between the training instance x and each of the instances y stored. Its purpose is to obtain the instance y most similar (y_{Max}) to instance x, in which the shortest distance is the greatest similarity. A potential similarity function may be found in Aha[1] and is also used by algorithms IB2 and IB3. IB1 stores all instances of a concept description, giving this classification a high cost of storage and processing. This IB1 limitation has been overcome by its successor, IB2.

2.2 The IB2 Algorithm

Aiming at the reduction of stored data, the IB2 stores only instances that were not classified correctly. As shown in Aha [1], the IB2 considerably decreased the need for storage, but became more sensitive than IB1 to noise presence in training data, due to the storage of incorrectly classified instances. Figure 2 illustrates the workings of IB2.

To circumvent the sensitivity to noise, an extension of IB2 was proposed. This extension has materialized itself in the IB3 algorithm.

2.3 The IB3 Algorithm

The IB3 algorithm aims to overcome the noise intolerance through a test that determines if a given instance will be relevant to future ratings or if such

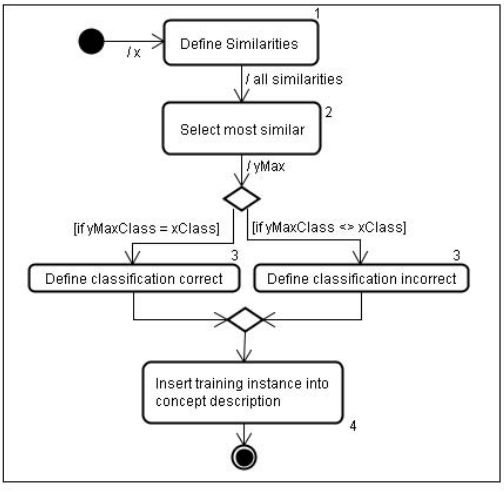


Fig. 1. Algorithm IB1 activity flow

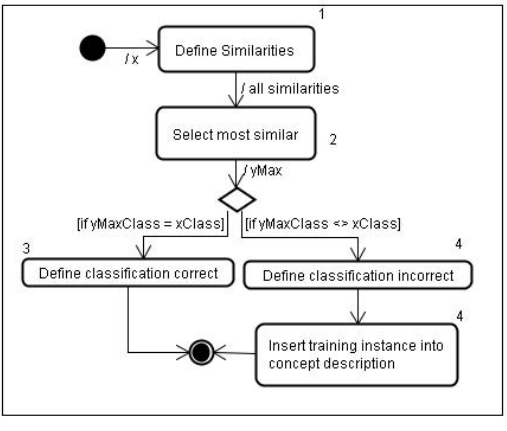


Fig. 2. Algorithm IB2 activity flow

proceedings represent only noise. To this end, a rating record with each instance stored is maintained. This record identifies the classification performance of the instance.

Figure 3 illustrates the operation of IB3. In step 2, the accepted instance with greater similarity (y_{max}) is selected. For an instance to be good or acceptable, its accuracy interval‘s lower endpoint should be greater than the class frequency interval‘s higher endpoint. If the accuracy interval‘s higher endpoint is less than the class frequency interval‘s lower endpoint, the instance is removed. Details about both accuracy interval and class frequency interval may be found

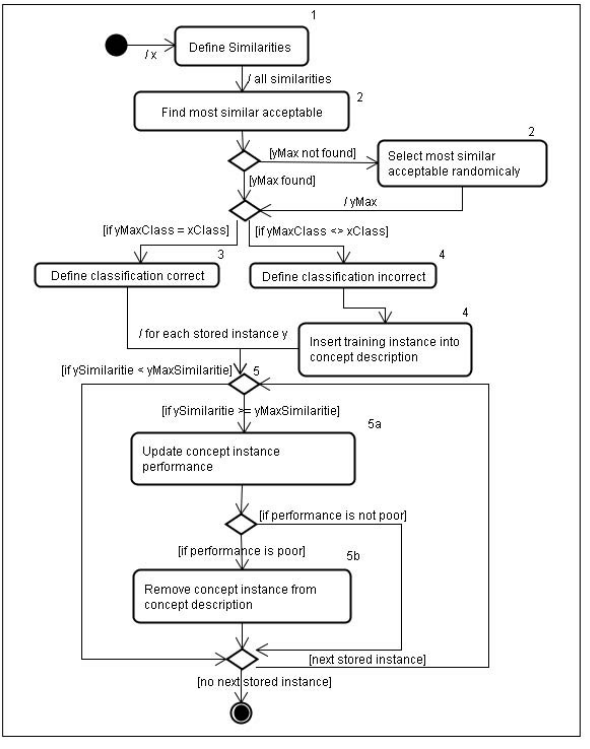


Fig. 3. Algorithm IB3 activity flow

in Aha[1]. In the fifth step, the similarity of the most similar acceptable instance y_{max} is compared to the similarity of each sample y stored in the concept description. Only samples with a similarity greater or equal to the similarity of y_{max} will have their performance changed, being subject to removal from the concept description.

3 Bayesian Network

Several methods based on theory probability have been introduced, among which one of the most used is the Bayesian Network. A Bayesian Network [6] is a directed acyclic graph formed by nodes and edges. A node is used to represent a variable, which can be discrete, continuous or propositional. The nodes are connected by links that represent the dependency between them. Each network node will have probabilistic relationships, in which the local probability value depends on the values of parent nodes.

A Bayesian Network can be constructed by using the knowledge of an expert, an automatic learning process through the analysis of previous data or a combination of both. The network construction is divided into two parts: structure learning and parameter learning. After the network construction, it is possible

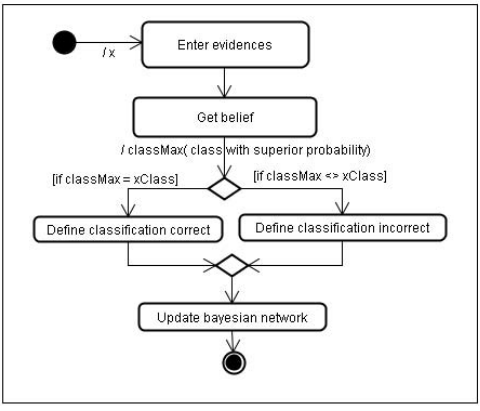


Fig. 4. Bayesian network activity flow

to perform probabilistic inferences. A probabilistic inference consists of drawing conclusions as new information or evidence is known. Probabilistic inferences generate beliefs for each network node, but do not change the knowledge base. Figure 4 ilustrates the operation of a bayesian network to classify a new instance x.

Bayesian learning has proved itself efficient in several studies [7, 14], allowing agents execute their processes with better negotiating skills. However, studies [12] also suggest that groups of learners are able to achieve superior performance to that of an isolated learner. The next section deals with an algorithm capable of coordinating a group of classifiers or experts. The goal here is twofold: to increase the classification performance and to efficiently detect the concept drifts.

4 Dynamic Weighted Majority

DWM [10] is an ensemble method for concept drift that keeps a group of expert agents (base classifiers), each implementing an algorithm. The operation of the DWM is detailed below:

1. for every new instance (i) received, the DWM:
   (a) starts weight prediction for each class with 0 (zero);
   (b) interacts with each expert of the group, sending the training instance: (i) each expert performs its local prediction; (ii) if the learning time did not exceed ($i \; mod \; p = 0$), the experts which made incorrect predictions have their weight decreased. The new weight is determined by $w(j) = w(j) * \beta$, where $w(j)$ is the weight of expert j and β is a constant with value between 0 and 1, used to decrease the experts' weight; (iii) the received prediction weight of the class instance is updated based on the weight of the expert in use.
2. after all the experts have made their classifications, the class with the highest weight will be considered in the overall prediction;

3. if the learning time did not exceed ($i \ mod \ p = 0$):
   (a) the weight of each expert is normalized, in which the highest weight should be 1;
   (b) experts with a weight below a certain minimum ($w[j] < \Theta$) are deleted, where Θ is a threshold for removal experts;
   (c) if the overall prediction is incorrect, a new expert is created with weight 1. It should be noted that the parameter p (a period of time) is required for environments with a high amount of noise.
4. Finally, the instance is sent to experts' training.

Being a generic coordination algorithm, the DWM allows the use of any learning algorithm. It is important to remember that the focus of this study is to compare the efficiency of the DWM setting with learning algorithm IB3, as demonstrated by Enembreck [5], with the efficiency of the DWM setting with Bayesian Networks. The developed environments and the experiments are detailed subsequently.

5 Test Environment

To evaluate the proposed study, software modules simulating a process of bilateral negotiation between a buyer agent and seller agent were defined. The seller agent starts the negotiation process with the publication of an interesting offer definition, and the buyer agent tries to create offers that are interesting to its opponent. Over time, the seller agent changes its interesting offer concept and the purchasing agent should automatically detect and adapt itself to this change.

Two settings were developed. First, the classification of offers and the detection of drift in the seller‘s concept are performed by a set of experts implementing the IB3 algorithm coordinated by DWM; afterward, the DWM coordinates a set of experts implementing Bayesian Networks. Each expert keeps an updated instance base, which represents the concept description.

The Table 1 gives an idea of the conceptual description of interesting offers for moderate drift. These concepts and the attributes used in the test environment were based on Enembreck [5], in which the reader may find details on the attributes and concepts in question.

The process begins by generating a set T of 50 test instances for each concept (c). These instances are used, one instance (i) at a time, to evaluate the accuracy of the system. The DWM, taking upon the role of process coordinator, verifies every test instance individually so each expert can make its own local prediction.

Table 1. Conceptual Description for Moderate Drift

ID	Description
1	(Delivery Delay = very low and Amount = very little)
2	(Delivery Delay = very low and Amount = little)
...	...
8	(Delivery Delay = very low and Amount = non ensured)

After all experts have made their prediction for a single instance of T, the DWM chooses the class with the highest weight as the overall prediction. This process is repeated until all test instances have been classified. System performance is calculated using Equation 1. TP (True Positives) and FN (False Negatives) are the number of interesting and non-interesting test instances correctly classified respectively and $|T|$ is the size of the test instances group.

$$accuracy(c, i) = \frac{TP + FN}{|T|} \times 100 \quad (1)$$

For each concept 50 training instances are also generated, so the weight of each expert is defined, experts with poor performance are removed, new experts are added to the group when necessary and finally, each training instance is sent to each of the expert, for training. IB3 experts perform classification using the similarity function described previously, while Bayesian Networks perform probabilistic inferences.

Bayesian Networks were built using the Netica API [3]. Each network node corresponds to a scalar variable; at the time of creation, the network received a uniform conditional probability distribution. Each new training instance corresponds to a new case, and its values are informed as findings or evidences. The probabilistic inference does not change the network's conditional probabilities, being executed to calculate the probability of an offer being interesting given the observed values of the attributes color, price, payment, amount and deliveryDelay. The network updating activity is restricted to the moment of training. On parameter learning, the framework used the Counting algorithm, due to its simplicity, fastness and suitability for situations where there is not much lack of information or uncertain events. For probabilistic inference, the framework used the Junction-Tree algorithm. The architecture of the Bayesian network is illustrated in Fig. 1.

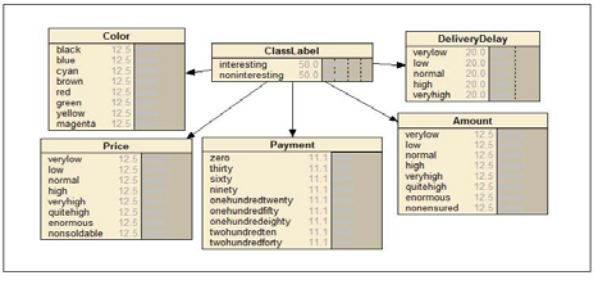


Fig. 5. Bayesian network architecture

6 Experiments

Experiments were performed using the following settings: DWM-IB3 and DWM-BN (Bayesian Network). Each setting was tested for moderate, gradual and abrubt concept drifts. The results presented in this section were generated from an average of 10 runs of each setting.

Tests carried out when the concept drift happened abruptly showed very similar results for the first two systems. Fig. 2 shows that the systems kept the lines that describe their performance very similar, 80% on average. Despite significant performance drops in the moments of concept drifts, both systems were able to restore performance in less than 10 iterations when the number of experts tended to stabilize or decrease (Concept 4 and Concept 5 for IB3 and Concept 5 to BN). A remarkable feature is the greater number of experts generated by the setting DWM-BN, as presented in Fig. 2.

Results obtained from data with moderate drift presented the performance of DWM-BN standing out slightly on the performance of DWM-IB3. Despite showing a higher performance drop, the DWM-BN could recover faster than the DWM-IB3. In this case the IB3 reacted more smoothly than it did when changes were abrupt, once a reasonable performance was reached, the stored instances presented good classification accuracy and were not modified so frequently. Fig. 3 shows that, same as with the previous setting, DWM-BN showed a larger amount of experts, while IB3 showed a tendency to stabilize itself.

Fig. 4 shows that during a gradual concept drift both DWM-IB3 and DWM-BN maintained an average performance of 80%. The performance drop in both settings was very similar and much less than those identified in Fig. 2 and Fig. 3, due the similarities among the produced concepts. The number of experts presented in Fig. 4 showed continued growth for the amount of generated data in both systems, tending to the beginning of stabilization in concept B.

The environments less affected by drifts and which managed to maintain a good performance with less sharp falls were the ones in which the concept drifted gradually and moderately. Moderate and gradual drifts also generated the smallest number of experts, once they are, theoretically, the easiest type of drifts to learn. The stabilization of the number of experts is more intense for the configuration DWM-IB3, because ensemble-based algorithms use the diversity of classifiers to cover different regions of the tuple space; when enough region is covered, some classifiers become useless, once they represent regions already covered by other classifiers. The DWM-BN setting allowed the presence of most

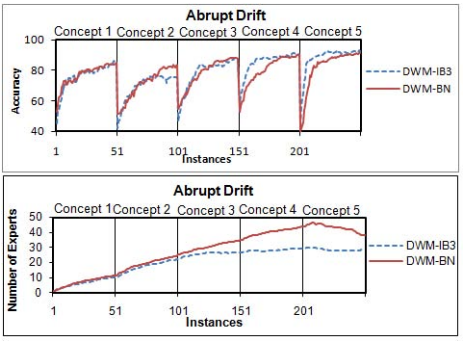


Fig. 6. Accuracy and number of experts with abrupt drift

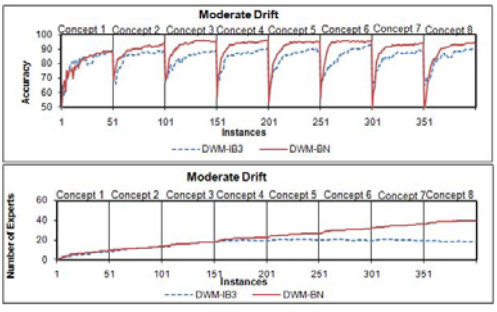


Fig. 7. Accuracy and number of experts with moderate drift

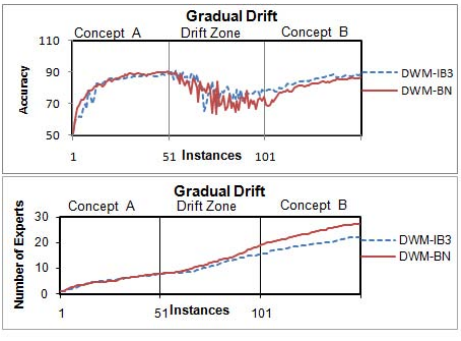


Fig. 8. Accuracy and number of experts with gradual drift

experts throughout the process. This behavior justifies the higher number of experts kept by DWM-BN, and, therefore, the need for more storage capacity and an execution time 60% higher on average than the other system.

7 Conclusions

The development of this work has allowed an efficient analysis of the IB3 and Bayesian Networks as learning algorithms coordinated by DWM in a scenario of automated negotiation. Three algorithms from the IBL family were presented (IB1, IB2 and IB3). The Bayesian Networks were incorporated into the study due to their high adaptability, as they allow new information to generate drifts in dependencies between concepts and also in the concepts themselves.

Aiming to increase the classification performance and efficiently detect concept drifts, the coordination model DWM was used in the experiments, contributing to increase the performance of learning algorithms. The experiments revolved around the settings DWM-IB3 and DWM-BN. In general, the performances of both systems were similar, detecting drifts quickly, adapting themselves by changing their concept description and recovering their performances

in a few iterations. Most of the time, both systems maintained similar performance lines, proving that both algorithms present a satisfactory performance in detecting drift when coordinated by DWM; however the setting DWM-BN requires massive memory resources and high processing time due to the need to update each network with the classified instance, while the updating is a low cost process for the setting DWM-IB3. These results strongly encourage the use of setting DWM-IB3 for the detection of different types of concept drifts in bilateral and multi-issue negotiations.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based Learning Algorithms. Machine Learning 6(1), 37–66 (1991)
2. Coehoorn, R.M., Jennings, N.R.: Learning an Opponent's Preferences to Make Effective Multi-Issue Negotiation Trade-Offs. In: Proc. of the Sixth International Conference on Electronic Commerce, pp. 59–68 (2004)
3. Norsys Software Corp., Netica-j manual (2009), http://www.norsys.com/neticaj/docs/netica#jman.pdf
4. Enembreck, F., Avila, B.C., Scalabrin, E.E., Barthès, J.-P.: Drifting Negotiations. In: Applied Artificial Intelligence, vol. 21(9), pp. 861–881. Taylor & Francis, Taylor (2007)
5. Enembreck, F., Tacla, C.A., Barthès, J.P.: Learning Negotiation Policies Using Ensemble-Based Drift Detection Techniques. International Journal of Artificial Intelligence Tools 18(2), 173–196 (2008)
6. Pearl, J.: Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann, San Mateo (1988)
7. Hindriks, K., Tykhonov, D.: Opponent modelling in automated multi-issue negotiation using bayesian learning. In: AAMAS 2008: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 331–338 (2008)
8. Kersten, G.E., Michalowski, W., Szpakowicz, S., Koperczak, Z.: Restructurable representations of negotiation. Manage. Sc. 37(10), 1269–1290 (1991)
9. Klinkenberg, R., Renz, I.: Adaptive Information Filtering: Learning in the Presence of Concept Drifts. In: ICML 1998, pp. 33–40 (1998)
10. Kolter, J.Z., Maloof, M.A.: Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift, p. 123. IEEE Computer Society, Los Alamitos (2003)
11. Littlestone, N., Warmuth, M.: The Weighted Majority algorithm. Information and Computation 108, 212–261 (1994)
12. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research 11, 169–198 (1999)
13. Ruggeri, F., Faltin, F., Kenett, R.: Bayesian Networks. In: Encyclopedia of Statistics in Quality & Reliability. Wiley & Sons, Chichester (2007)
14. Zeng, D., Sycara, K.: Bayesian learning in negotiation. International Journal of Human-Computer Studies 48(1), 125–141 (1998)

A Logic Based Algorithm for Solving Probabilistic Satisfiability*

Marcelo Finger** and Glauber De Bona

Department of Computer Science (IME-USP)
Institute of Mathematics and Statistics
University of Sao Paulo — Brazil
{mfinger,debona}@ime.usp.br

Abstract. This paper presents a study of the relationship between probabilistic reasoning and deductive reasoning, in propositional format. We propose an algorithm to solve probabilistic satisfiability (PSAT) based on the study of its logical properties. Each iteration of the algorithm generates in polynomial time a classical (non-probabilistic) formula that is submitted to a SAT-oracle. This strategy is a Turing reduction of PSAT into SAT. We demonstrate the correctness and termination of the algorithm.

1 Introduction

Probabilistic satisfiability (PSAT) is a problem that allows for the joint application of deductive and probabilistic reasoning; this form of reasoning can be performed without any a priori assumptions of statistical independence. Such reasoning capabilities may be very useful in modeling complex systems, where the statistical interdependence between the components is not known.

PSAT original formulation is credited to George Boole and dates back to 1854 [2]. The problem has since been independently rediscovered several times (see [9,10]) until it was introduced to the Computer Science and AI community by Nilsson [13] and was shown to be an NP-complete problem, even for cases where the corresponding classical satisfiability is known to be in PTIME [6]. Initial attempts to implement a solution led to the belief that the problem appeared to be very hard [14].

The PSAT problem is formulated in terms of a linear algebraic problem of exponential size. The vast majority, if not all, of algorithms for PSAT solving in the literature are based on linear programming techniques, such as column generation and matrix decomposition; see [10] for a good summary. There were also works covering extensions of PSAT with imprecise probabilities [11], and the discovery of some tractable fragments of PSAT [1], after which very little was published on algorithms for PSAT.

* This work was supported by Fapesp Thematic Project 2008/03995-5 (LOGPROB).
** Partially supported by CNPq grant PQ 304607/2007-0.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 453–462, 2010.

In the beginning of this century, very efficient solvers that dealt with the classical (non-probabilistic) satisfiability problem (SAT) became available [12,7]. The SAT problem is a well studied problem, with known hard and easy instances and a phase-transition complexity distribution [5]. Efficient SAT-solvers enabled the solution of other NP-complete problems by translating them to a SAT instance and applying the solvers. Cook's theorem guarantees that there exists a polynomial-time translation from any NP-complete problem into SAT. However, no such translation from PSAT to SAT is found in the literature.

The aim of this paper is to study the logical properties of PSAT so as to present a *Turing reduction* of PSAT into SAT. A Turing reduction of PSAT into SAT is an algorithm to decide PSAT that employs a SAT-oracle [15].

The rest of the paper develops as follows. Section 2 presents the PSAT problem and proposes an *atomic normal form* to deal with it. The logical properties of PSAT are studied via a probabilistic entailment relation in Section 3, which allows for the presentation of a PSAT solving algorithm. Sections 4 and 5 demonstrate its correctness and that every step of the algorithm can be performed in polynomial time. Termination of the algorithm is shown in Section 6. Proofs have been omitted due to space restrictions.

2 The PSAT Problem

Probabilistic satisfiability consists in determining if the following configuration is consistent: a finite set of k classical propositional formulas $\alpha_1, \ldots, \alpha_k$ defined on n logical variables $\mathcal{P} = \{x_1, \ldots, x_n\}$, restricted by a set of probabilities imposed on the truth of these formulas, $p_1, \ldots, p_k$, such that $P(\alpha_i) = p_i, 1 \leq i \leq k$.

Formally, consider $\mathcal{L}$ the set of all propositional formulas, which may or may not be in clausal form. A propositional valuation v is initially defined over propositional variables, $v : \mathcal{P} \to \{0,1\}$ and then is extended, as usual, to all formulas, $v : \mathcal{L} \to \{0,1\}$. Consider the set V of 2^n propositional valuations $v_1, \ldots, v_{2^n}$ over the n variables. A *probability distribution over propositional valuations* $\pi : V \to [0,1]$ is a function that maps every valuation to a value in the real interval $[0,1]$ such that $\sum_{i=1}^{2^n} \pi(v_i) = 1$. The probability of a formula α according to distribution π is given by $P_\pi(\alpha) = \sum\{\pi(v_i) | v_i(\alpha) = 1\}$.

Define a $k \times 2^n$ matrix $A = [a_{ij}]$ such that $a_{ij} = v_j(\alpha_i)$. The *probabilistic satisfiability problem* is to decide if there exists a probability vector π of dimension 2^n such that obeys the *PSAT restrictions*:

$$\begin{aligned} A\pi &= p \\ \sum \pi_i &= 1 \\ \pi &\geq 0 \end{aligned} \tag{1}$$

The last two conditions force π to be a probability distribution. The condition $\sum \pi_i = 1$ is inserted as the first line of matrix A, that becomes a $(k+1) \times 2^n$ matrix, and $p_{(k+1)\times 1}$ is now a vector with $p_0 = 1$. A *PSAT instance* is a set $\Sigma = \{P(\alpha_i) = p_i | 1 \leq i \leq k\}$, which is *satisfiable* iff its associated PSAT restrictions (1) have a solution. If π is a solution to (1) we say that π satisfies Σ.

An important result of [6] guarantees that a solvable PSAT instance has a "small" witness.

Proposition 1 ([6]). *A PSAT instance* $\Sigma = \{P(\alpha_i) = p_i | 1 \leq i \leq k\}$ *has a solution iff there is a set of* $k+1$ *columns of* A *such that the resulting* $(k+1) \times (k+1)$ *system has a positive solution.*

The solution given by Proposition 1 serves as an NP-certificate for this instance, so PSAT is in NP. Furthermore, as *propositional satisfiability* (SAT) is a subproblem obtained when all $p_i = 1$, PSAT is NP-hard. So PSAT is NP-complete.

It follows from the Cook-Levin Theorem [3] that there must be a polynomial time reduction from PSAT to SAT. However, finding an efficient such reduction is neither obvious nor easy at all. But considering the fast advances in the construction of SAT solvers [12], it is a worthwhile task for two reasons. On the one hand, there is the basis for a general and relatively efficient implementation of PSAT which may render it useful in practical applications of moderate size, whose scale may increase with the advances in SAT solving. On the other hand, there is the understanding one gets from the relationship between probabilities and logic.

On the latter topic, one must add that the vast majority of PSAT solving initiatives, as surveyed in [10], concentrated in solutions based on variations of linear programming. But here we are interested in exploring the same problem via logical techniques. As PSAT poses no form of a priori probabilistic independence assumption, it is actually a form of probabilistic reasoning that is not very much explored in the literature.

First, some notation. If A is an $(k+1) \times (k+1)$ matrix, A^j represents its j-th column, and if $b_{(k+1)\times 1}$ is a column, $A[j := b]$ represents the matrix obtained by substituting b for A^j; $|A|$ is A's determinant.

2.1 The Atomic Normal Form

We say that a PSAT instance $\Sigma = \{P(\alpha_i) = p_i | 1 \leq i \leq l\}$ is in the *atomic normal form* if it can be partitioned in two sets, (Γ, Ψ), where $\Gamma = \{P(\alpha_i) = 1 | 1 \leq i \leq m\}$ and $\Psi = \{P(y_i) = p_i | y_i$ is an atom and $1 \leq i \leq k\}$, with $0 < p_i < 1$, where $l = m + k$. The partition Γ is the SAT part of the normal form, usually represented only as a set of propositional formulas and Ψ is the *atomic probability assignment* part. An atomic PSAT instance is a PSAT instance in atomic normal form.

Theorem 1 (Atomic Normal Form). *Let* $\Sigma = \{P(\alpha_i) = p_i | 1 \leq i \leq k\}$ *be a PSAT instance. Then there exists an atomic PSAT instance* (Γ, Ψ) *such that* Σ *is a satisfiable iff* (Γ, Ψ) *is; the atomic instance can be built from* Σ *in polynomial time.*

The atomic normal form allows us to see a PSAT instance (Γ, Ψ) as an interaction between a probability problem (represented by Ψ) and a SAT instance Γ. Solutions to the instance can be seen as solutions to Ψ constrained by the SAT

instance Γ. First, note that $\Psi = \{P(y_i) = p_i | 1 \leq i \leq k\}$ is a PSAT instance such that the columns of its corresponding matrix Y_Ψ are valuations over $y_1, \ldots, y_k$. However, for one solution to this instance to be a solution to the atomic instance (Γ, Ψ), this solution must be further constrained.

We say that a valuation v over $y_1, \ldots, y_k$ is consistent with a SAT instance Γ over variables $y_1, \ldots, y_k, x_1, \ldots, x_n$ if there is an extension of v over $y_1, \ldots, y_k, x_1, \ldots, x_n$ such that $v(S) = 1$ for every $S \in \Gamma$.

Lemma 1. *Let $\Psi = \{P(y_i) = p_i | 1 \leq i \leq k\}$. An atomic instance (Γ, Ψ) is satisfiable iff there is a $(k+1) \times k'$-matrix A_Ψ verifying the PSAT restrictions (1), $k' \leq k+1$, such that every column of A_Ψ is a valuation over $y_1, \ldots, y_k$ consistent with Γ.*

By Lemma 1, an atomic PSAT instance (Γ, Ψ) has a solution iff there is a matrix A satisfying conditions (2) below such that if $\pi_j > 0$ then $a_{1,j}, \ldots, a_{k,j}$ are Γ-consistent valuations for $y_1, \ldots, y_k$, $1 \leq j \leq k+1$:

$$\begin{bmatrix} 1 & \cdots & 1 \\ a_{1,1} & \cdots & a_{1,k+1} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \cdots & a_{k,k+1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ \vdots \\ p_k \end{bmatrix} \tag{2}$$

$\pi_j \geq 0$, A is non-singular,
$a_{i,j} \in \{0, 1\}$, $1 \leq i \leq k$, $1 \leq j \leq k+1$

We will further assume, without loss of generality, that the input probabilities $p_1, \ldots, p_k$ are in decreasing order.

2.2 Relaxed PSAT

Given an atomic PSAT instance (Γ, Ψ) its associated *relaxed PSAT instance* is obtained by ignoring the restriction Γ, namely the instance $(\varnothing, \Psi)$.

Lemma 2. *Every relaxed PSAT instance is satisfiable.*

The matrix A_0 is called the *relaxed matrix of order* $k+1$ and consists of 1's on the diagonal and above it, and 0's below it. We assume we have a function *relaxedMatrixOfOrder*(k) that generates it.

The basic idea of the PSAT decision using a SAT-oracle is to start with a solution for the relaxed instance and try to substitute the columns of A_0 that are not consistent with Γ by others that are. The generation of these columns will be solutions to auxiliary SAT instances. If some auxiliary instance is unsatisfiable, so is the PSAT instance.

But before we can explain how the auxiliary SAT instances work, we need to introduce a few logical and algebraic tools.

3 A Probabilistic Entailment Relation

We propose a *probabilistic entailment relation*, $\mid\!\approx$, between an atomic probability assignment Ψ and a formula α, such that $\Psi \mid\!\approx \alpha$ iff the atomic PSAT instance $(\{\neg\alpha\}, \Psi)$ is (probabilistically) unsatisfiable. The intuition is that Ψ forces the $P(\alpha) > 0$; in fact, if $(\{\neg\alpha\}, \Psi)$ is unsatisfiable, this means that $P(\alpha) = 0$ is inconsistent with Ψ, but since $P(\alpha) \geq 0$ for any formula, it follows that $\Psi \mid\!\approx \alpha$ implies $P(\alpha) > 0$ for any probability distribution that satisfies Ψ.

Probabilistic entailment plays a special role in the study of probabilistic satisfiability, in the following sense.

Lemma 3. *Consider an atomic PSAT instance $\Sigma = (\Gamma, \Psi)$. Σ is satisfiable iff $\Gamma \cup \{\alpha\}$ is classically satisfiable for every α such that $\Psi \mid\!\approx \alpha$.*

Corollary 1. *$\Sigma = (\Gamma, \Psi)$ is (probabilistically) unsatisfiable iff there is a γ such that $\Psi \mid\!\approx \gamma$ and $\Gamma \models \neg\gamma$.*

Consider a PSAT solution of the form (2) and $\Gamma = \Gamma(y_1, \ldots, y_k; x_1, \ldots, x_n) = \Gamma_i(y; x)$. If we want to represent each column $A^j = \{a_{1,j}, \ldots, a_{k,j}\}$ as a set of variables $y^j = \{y_1^j, \ldots, y_k^j\}$, we have that $\Gamma(y^j; x^j)$, $1 \leq j \leq k+1$, have all disjoint sets of variables and have all to be jointly satisfiable. If we have a set of classical conditions $C = \{\alpha_1, \ldots, \alpha_c\}$ such that $\Psi \mid\!\approx \alpha_i$, $1 \leq i \leq c$, no matter how big the number of conditions c is, all conditions of C have to be satisfied by one of at most $k+1$ valuations, which are columns of the matrix A. Therefore, $\alpha_i(y^1; x^1) \vee \ldots \alpha_i(y^{k+1}; x^{k+1})$ must hold for each $\alpha_i \in C$. Furthermore, all these conditions have to be jointly satisfiable, as there are only $k+1$ valuations allowed in a solution of the form (2). We have thus proved the following.

Lemma 4. *Let C be a set of Ψ-entailed formulas. Then the following set of formulas must be jointly satisfiable:*

$$\begin{array}{l} \{\Gamma(y^j; x^j) | 1 \leq j \leq k+1\} \cup \\ \{\alpha(y^1; x^1) \vee \ldots \vee \alpha(y^{k+1}; x^{k+1}) | \alpha \in C\} \end{array} \tag{3}$$

We assume there is a function *satFormulaFromConditionSet*(C) that given a set of Ψ-entailed formulas, generates a formula φ of the format (3) to be sent to a SAT oracle. If φ is unsatisfiable, by Corollary 1 the initial PSAT instance is also unsatisfiable. Otherwise it outputs a valuation v from which we can assemble a $(k+1) \times (k+1)$ matrix B extracting the value v attributes to y^j, $1 \leq j \leq k+1$; the first line of B is all 1's. We assume there is a function *assembleMatrix*(v) that constructs such matrix.

Note that B may not satisfy conditions (2) as, for instance, it may be singular in the case that all conditions of C are satisfied by a $k' < k+1$ columns, so that $v(y^{k''}) = v(y^{k'})$ for $k'' > k'$. So we have to merge the columns of B with those of A to generate a new matrix satisfying conditions (2). We are now in a position to present an outline of the PSAT-solver algorithm.

3.1 Algorithm

Based on Lemma 4 we are in a position to present the outline of the algorithm that decides a PSAT instance employing a number of applications of the SAT problem, which is presented in Algorithm 3.1.

Algorithm 3.1. PSAT via SAT

Input: An atomic normal form PSAT instance $\Sigma = (\Gamma, \Psi = \{P(y_i) = p_i | 1 \leq i \leq k\})$.
Output: No, if Σ is unsatisfiable. Or a solution matrix A satisfying (2).

```
1:  i := 0
2:  A_0 := relaxedMatrixOfOrder(k + 1)
3:  C_0 := ∅ //No initial conditions
4:  while A_i is not a solution for (Γ, Ψ) do
5:     C_{i+1} = C_i ∪ conditionsFromMatrix(A_i)
6:     φ := satFormulaFromConditionSet(C_{i+1})
7:     if φ is SAT-solvable with valuation v then
8:        B := assembleMatrix(v)
9:        A_{i+1} = merge(A_i, B) //Invariant: A_{i+1} satisfies (2)
10:       increment i
11:    else
12:       return No
13:    end if
14: end while
15: return A_i
```

So, the basic idea is to start every iteration with a matrix A_i satisfying (2), use it to generate a set of Ψ-entailed formulas, generate a formula φ from it and submit it to a SAT-oracle; if φ is unsatisfiable, so is the initial instance Σ. Otherwise, obtain a matrix B satisfying all conditions of φ and merge it with A_i, generating A_{i+1} satisfying (2) and check if its a solution. If it is, return A_{i+1}; otherwise, repeat this process.

To check for a solution (line 4), we inspect the vector π in the invariant (2). A solution is found when for all components $\pi_j > 0$, column A_i^j represents a Γ-consistent valuation. If $\pi_j = 0$, we need not enforce A_i^j to be a Γ-consistent valuation, as any other column can be substituted for it; this represents the case where a solution for (1) can be found with rank less than $k + 1$.

To complete the description of Algorithm 3.1, we still have to clarify the following points:

- Line 5: how to obtain a set of Ψ-entailed formulas from a matrix satisfying (2).
- Line 9: how to merge a matrix A satisfying (2) with another, generating a third one that still satisfies (2), keeping the invariant a line 9.
- Line 4: how the loop halts in a finite number of steps.

4 Linear Algebra and Logic

In this section we examine some relations between Linear Algebra and Logic. In particular, we obtain Ψ-entailed formulas from a matrix satisfying (2).

4.1 Linear Restrictions as Formulas

A linear restriction over variables $x_1, \ldots, x_k$ is a condition of the form

$$a_1 \cdot x_1 + \cdots a_k \cdot x_k \quad op \quad c \tag{4}$$

where $a_1, \ldots, a_k, c \in \mathbb{Q}$ and $op \in \{<, \leq, >, \geq, =, \neq\}$. If we further restrict the variables x_i such that $x_i \in \{0, 1\}$, $1 \leq i \leq k$, a linear restriction LR of the form (4) can be seen as a formula, in the sense that a valuation v such that $v(x_i) \in \{0, 1\}$ *satisfies* LR if the restriction is verified when we substitute $v(x_i)$ for x_i in (4). This view of linear-restrictions-as-formulas was developed in the area of linear programming [8], and has been used in pseudo-boolean inference methods [16,4].

Proposition 2. *Every linear restriction LR can be represented by a set of clauses Δ_{LR} such that LR is satisfiable iff Δ_{LR} is; the number of atoms and clauses in Δ is bound by polynomial in k.*

For example, the linear restriction

$$2x_1 - x_2 + x_3 \geq 2$$

when restricted to $x_1, x_2, x_3 \in \{0, 1\}$ can be seen as representing the clausal form formula $x_1 \wedge (\neg x_2 \vee x_3)$.

Proposition 2 allows us to deal with linear restrictions as a formula, without worrying about an exponential explosion, as they can be brought to clausal normal form in polynomial time.

4.2 Logical Restrictions for Column Substitution

We now study the logical conditions for substituting columns of a given matrix that satisfy the invariant condition (2). Given a matrix A satisfying (2), we will construct a matrix $C_{k+1 \times k+1} = [c_{i,j}]$ of logical conditions.

Recall that A^j represents A's j-th column, and if $b_{(k+1)\times 1}$ is a column, $A[j := b]$ represents the matrix obtained by substituting b for A^j.

Let $y = [1\ y_1\ \ldots\ y_k]'$ be a column of variables. If A satisfies (2), and we want to substitute its j-th column by $b = [1\ b_1\ \ldots\ b_k]'$, to keep (2) invariant, $A[j := b] \cdot \pi = p$, then necessarily, $|A[j := b]| \neq 0$. This condition may be stated syntactically by the linear-restriction-as-formula $|A[j := y]| \neq 0$, where the logical restrictions on y represent the algebraic restrictions on b; in fact, this restriction is linear for *only one column* is being substituted by variables. This may already be expressed as a logical formula. However, more expressivity can be obtained from *Cramer's Rule*, according to which:

$$\pi_j = \frac{|A[j := y][j := p]|}{|A[j := y]|} \geq 0 \tag{5}$$

Note that $A[j := y][j := p] = A[j := p]$ contains no variables. Let $\text{sign}(x) = -1$, if $x < 0$, and $+1$ otherwise. If $|A[j := p]| \neq 0$, then $\pi_j > 0$, so (5) yields

$$sign(|A[j := p]|) \cdot |A[j := y]| > 0 \tag{6}$$

which is a linear restriction. In the matrix of logical conditions C we make the diagonal elements represent these conditions. Let $A' = A[j := p]$, then

$$c_{j,j} = \begin{cases} sign(|A'|) \cdot |A[j := y]| > 0, & \text{if } |A'| \neq 0 \\ |A[j := y]| \neq 0, & \text{if } |A'| = 0 \end{cases} \tag{7}$$

Note that the $c_{j,j}$ enforce that the column that substitutes A^j, that is non-singular, will make the resulting matrix also non-singular. The other elements of matrix C are also obtained using Cramer's rule. Fix column j and consider a column $i \neq j$; we have

$$\pi_i = \frac{|A[j := y][i := p]|}{|A[j := y]|} \geq 0 \tag{8}$$

As $c_{j,j}$ enforces $|A[j := y]| \neq 0$. If we further have that $|A[j := p]| \neq 0$, (6) and (8) yield that

$$sign(|A[j := p]|) \cdot |A[j := y][i := p]| \geq 0,$$

which is a linear restriction and can be expressed by a formula $\chi_{i,j}$. When $|A[j := p]| = 0$ we need to consider two cases, namely $A[j := y] > 0$ and $A[j := y] < 0$; as both are linear restrictions that can be expressed as formulas, let us abbreviate those formulas as ψ_j and ξ_j, respectively. Now consider the following linear conditions, $|A[j := y][i := p]| \geq 0$ and $|A[j := y][i := p]| \leq 0$ and let us abbreviate the formulas that encode those conditions as $\psi_{i,j}$ and $\xi_{i,j}$; it is clear that when ψ_j holds, we want $\psi_{i,j}$ to hold, and when ξ_j holds we want $\xi_{i,j}$ to hold. In the matrix of logical conditions C we make the non-diagonal elements represent these conditions. Let $A' = A[j := p]$ and $i \neq j$, then

$$c_{i,j} = \begin{cases} \chi_{i,j}, & \text{if } |A'| \neq 0 \\ (\psi_j \wedge \psi_{i,j}) \vee (\xi_j \wedge \xi_{i,j}), & \text{if } |A'| = 0 \end{cases} \tag{9}$$

The elements of the condition matrix $C = [c_{i,j}]$, $1 \leq i, j \leq k + 1$, are given by (7) and (9) and are all logical formulas. Each column C^j is constructed concentrating on its corresponding column A^j and the restrictions on π_i, $1 \leq i \leq k + 1$. The following is a central property of the elements of matrix C.

Lemma 5. *Consider a matrix of conditions $C = [c_{i,j}]$ constructed as above based on a matrix A satisfying (2), $A \cdot \pi = p$. If $\pi_j > 0$ then $\Psi \approx c_{i,j}$, $1 \leq i, j \leq k + 1$.*

Lemma 5 states that the elements of the condition matrix C are conditions of the form needed in line 5 of PSAT-solving Algorithm 3.1. So we define

$$conditionsFromMatrix(A) = \{c_{i,j} \in C\} \cup haltingCondition(A) \ . \qquad (10)$$

where $haltingCondition(A)$ will be developed in Section 6.

5 Merging

Merging consists taking a matrix A satisfying the invariant conditions (2) and a matrix B whose columns represent Γ-consistent valuations and generating a new matrix A' that incorporate "some" of B columns such that A' satisfies the invariant conditions (2). The following lemma tells us how to merge a single column.

Lemma 6. *Let A be a matrix satisfying the invariant condition (2) and let $b = [1\ b_1\ \cdots\ b_k]'$ be a column such that $b_i\{0,1\}, 1 \leq i \leq k$. Then there always exists a column j such that $A[j := b]$ satisfies the invariant conditions (2).*

The proof of Lemma 6 gives a procedure to merge a single column b with a given matrix A preserving invariant condition (2). Consider matrix $B = [b_1 \dots b_{k+1}]$ as a sequence of columns. We define the $merge(A, B)$, used in line 9 in Algorithm 3.1, as the successive merging of each column b_i. As each merging step, by Lemma 6, preserves the invariant condition (2), so does the output of $merge(A, B)$.

As noted above, the substituted column always exists in the merge process, but to ensure termination we assume that the merge procedure always chooses to substitute a column that is not Γ-consistent, if one is available.

6 Termination

To guarantee termination the halting formula will force at least one Γ-inconsistent formula to be substituted at each iteration of Algorithm 3.1.

Let A be a $\{0,1\}$-matrix satisfying invariant conditions (2), and let $C = [c_{i,j}]$ be its corresponding condition matrix as in Section 4.2. Then any column y substituting A^j has to satisfy all the conditions in column C^j; let $d_{i,j}(y)$ be defined as

$$d_{i,j}(y) = c_{i,j}(y) \wedge c_{j,j}(y) \ .$$

As $c_{j,j}$ makes $|A[j := y]| \neq 0$, $d_{i,j}(y)$ guarantees that $A[j := y]$ is non-singular. Considering the merging method of Section 5, if after ℓ substitutions every $d_{i,j}$, $1 \leq i \leq k+1$, was satisfying by a substituting column, then column j has been substituted in the merging process. The halting formula guarantees that j is a Γ-inconsistent column of A.

Suppose there are $\ell - 1$ Γ-consistent columns in A, and consider the set $S = \{j | A^j$ is a Γ-inconsistent columns with $\pi_j > 0\}$. Then the halting condition is

$$haltingCondition(A) = \bigvee_{j \in S} \bigwedge_{i=1}^{k+1} d_{i,j}(y^1) \vee \ldots \vee d_{i,j}(y^\ell) \ . \qquad (11)$$

Theorem 2. *Algorithm 3.1 terminates.*

7 Conclusions and Future Work

We have presented an algorithm that decides PSAT using a SAT-oracle, a theoretical work with potential practical applications.

Future work contemplates an implementation of a PSAT solver based on the algorithm presented here, which will employ an existing SAT-solver (such as zchaff [12]) as an implementation of the SAT oracle. We plan to make this PSAT-solver available as open-source software.

References

1. Andersen, K., Pretolani, D.: Easy cases of probabilistic satisfiability. Annals of Mathematics in Artificial Intelligence 33(1), 69–91 (2001)
2. Boole, G.: An Investigation on the Laws of Thought. Macmillan, London (1854)
3. Cook, S.A.: The complexity of theorem-proving procedures. In: Third Annual ACM Symposium on Theory of Computing (STOC), pp. 151–158. ACM, New York (1971)
4. Dixon, H.E., Ginsberg, M.L.: Inference methods for a pseudo-boolean satisfiability solver. In: AAAI/IAAI, pp. 635–640 (2002)
5. Gent, I.P., Walsh, T.: The SAT phase transition. In: ECAI 1994 – Proceedings of the Eleventh European Conference on Artificial Intelligence, pp. 105–109 (1994)
6. Georgakopoulos, G., Kavvadias, D., Papadimitriou, C.H.: Probabilistic satisfiability. Journal of Complexity 4(1), 1–11 (1988)
7. Goldberg, E., Novikov, Y.: Berkmin: A Fast and Robust SAT Solver. In: Design Automation and Test in Europe (DATE 2002), pp. 142–149 (2002)
8. Gomory, R.: An algorithm for integer solutions to linear programs. In: Recent Advances in Mathematical Programming, pp. 69–302. McGraw-Hill, New York (1963)
9. Hailperin, T.: Prob. logic. N. D. Journal of Formal Logic 25(3), 198–212 (1984)
10. Hansen, P., Jaumard, B.: Probabilistic satisability. Technical Report, Les Cahiers du GERAD G-96-31, École Polytechique de Montréal (1996)
11. Hansen, P., et al.: Probabilistic satisfiability with imprecise probabilities. In: 1st Int. Symposium on Imprecise Probabilities and Their Applications, pp. 165–174 (1999)
12. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an Efficient SAT Solver. In: Proceedings of the 38th Design Automation Conference (DAC 2001), pp. 530–535 (2001)
13. Nilsson, N.: Probabilistic logic. Artificial Intelligence 28(1), 71–87 (1986)
14. Nilsson, N.: Probabilistic logic revisited. Artificial Intelligence 59(1-2), 39–42 (1993)
15. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)
16. Warners, J.P.: A linear-time transformation of linear inequalities into conjunctive normal form. Inf. Process. Lett. 68(2), 63–69 (1998)

On-Line Viscosity Virtual Sensor for Optimizing the Combustion in Power Plants

Pablo H. Ibargüengoytia and Miguel Angel Delgadillo

Instituto de Investigaciones Eléctricas
Av. Reforma 113, Palmira
Cuernavaca, Mor., 62490, México
{pibar,madv}@iie.org.mx

Abstract. Fuel oil viscosity is an important parameter in the control of combustion in power plants. If the viscosity is optimal at the entrance of the boiler, then the combustion is optimal causing a minimum of contamination and a maximum of efficiency. Hardware viscosimeters are expensive and difficult to operate. Laboratory analyses calculate the viscosity based on chemical analysis but not in real time. This paper describes the development of a virtual sensor that utilizes artificial intelligence (AI) techniques for the construction of models. The models are used to estimate the viscosity based on related measurements concerning the combustion in a power plant. A probabilistic model is constructed using automatic learning algorithms and an analytical model is defined using physical principles and chemical analysis. Sensor fusion is applied to estimate the on-line value of the fuel viscosity. The virtual sensor is being installed in the Tuxpan power plant in Veracruz, Mexico.

Keywords: virtual sensors, fuel oil viscosity, Bayesian networks, learning.

1 Introduction

In Mexico, 42% of the electric power generation is based on fossil fuel [2]. The fuel is burned in boilers in order to produce steam that move turbines and finally they move electric generators. Generally, the fuel oil that is delivered to the thermoelectric power plants is the last sub-product of the oil refinement. The quality and characteristics of several fuels vary on different aspects. Burners in the boiler require fuel in a specific characteristic in order to be burn properly. The fuel is atomized and passed to the combustion chambers. If the droplet of fuel is too small, the combustion is not complete and generates contaminant emissions. On the other hand, if the droplet is too big, it doesn't burn completely and the combustion generates contamination and produces low efficiency.

The main characteristic of the fuel is the viscosity. Viscosity is a property of the matter that offers a resistance to flow. Hardware viscosimeters are expensive and difficult to operate and to maintain. Imagine to measure the flow of liquids that behave almost like a solid. The second option to measure the viscosity is

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 463–472, 2010.

chemical analysis in laboratory. This option is used every time a new supply of fuel arrives to a power plant. However, this procedure is off line and takes more than an hour. This paper presents the development of a viscosity virtual sensor that *estimates* the value of the viscosity given related measurements. In a previous paper [5], it was shown the basic principles of the design of a virtual sensor using Bayesian networks. In that project, the probabilistic model was constructed with chemical composition information obtained from laboratory analysis. However, this estimation is not on-line and it cannot be used for real time combustion control.

This paper describes the construction of a viscosity virtual sensor that works on-line in a power plant. The Federal Commission of Electricity (CFE) in Mexico is supporting this project to install a first prototype in a selected power plant, and later, replicate this sensor in all plants that contain boilers.

The challenges of this project are described in this paper and the proposed solution is explained. Two kinds of contributions are provided in this project. First are the technological challenges and second are the practical contributions in the combustion control of traditional power plants. The technological contributions include the construction of the probabilistic model using learning algorithms, and the use of analytic model using physical principles. The fusion of these inferences is also presented. The project is based on the hypothesis that there exist a number of normal variables that are probabilistically related with the viscosity and can be used to estimate it. The practical contribution consists in the capacity of estimate the fuel viscosity in order to control the temperature and obtain an optimal atomization of the fuel and hence, an optimal combustion. Besides, once that the first virtual sensor is completely installed, it can be replicated in all the boilers of power plants in Mexico.

The paper is organized as follows. The next section presents the system architecture and establishes the problems that are solved. Section 3 describes the process of acquiring data for constructing the models. Section 4 the pre-processing of all data, including the normalization and the discretization. Next, section 5 explains the complete process of automatic learning in order to construct the probabilistic model. Section 6 explains the analytic model. Finally, section 7 discusses the results obtained in the installation of the virtual sensor in the Tuxpan central in Mexico.

2 System Architecture

Figure 1 shows a schematic diagram of the viscosity virtual sensor project.

The virtual sensor design starts in the selection and acquisition of the related signals from the plant. The hypothesis is that the related signals may come from: before, during and after the combustion. In the selected plant there is a hardware viscosimenter that is used to compare the process signals with the measured viscosity. Thus, a huge historical data set is acquired with measures every 5 seconds during several days. In a first attempt, several variables were sampled. This represented an enormous number of measurements.

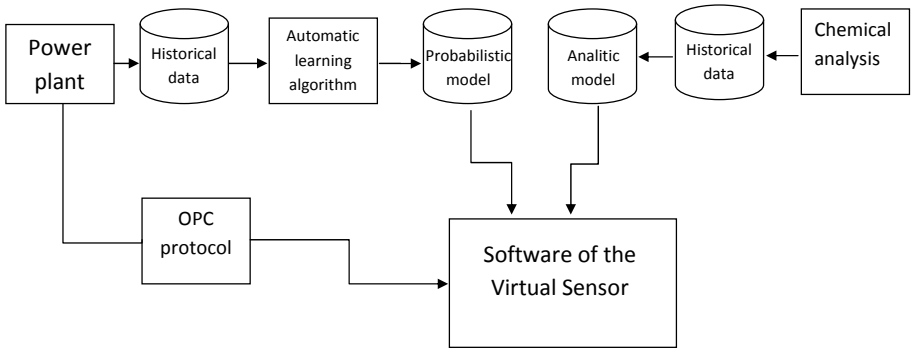


Fig. 1. Block diagram of the viscosity virtual sensor

The data set is cleaned, normalized and discretized to be ready for the learning algorithms. With the learning algorithm, a probabilistic model is constructed, based on Bayesian networks[6]. On the right side of Fig. 1, physical principles and chemical laboratory data allows constructing an analytic model. Chemical composition of different qualities of fuel were consulted and utilized in the analytic model. The construction of both models, the probabilistic and the analytic, are carried out off-line. They are later utilized in the virtual sensor software. This computer program opens the models and reads real time data. Each model infers the viscosity value and then, the fusion module calculates the final viscosity value. The next sections describe every block in Fig. 1.

3 Data Acquisition

The first problem was the selection of related signals. Diagrams with all the equipment and the instrumentation installed in the plan were provided (DTIs) and a full explanation of the performance of units 1 and 2 of the Tuxpan Thermoelectric power plant.

Tuxpan is a 6 units power plant located at north of the state of Veracruz, in the Gulf of Mexico littoral. This plant was selected since it has installed viscosimeters in units 1 and 2. This instrument is needed to acquire historical data including the viscosity in order to find the relationship between viscosity and other signals.

In a first attempt, 32 variables were selected for constructing the probabilistic model. However, revising the behavior of each signal with respect to the viscosity, and consulting the experts in combustion, only a few variables remain. Table 1 describes the ID and the description of the variables selected. Besides the variables extracted from the power plant data base, some other were calculated. The first one is the Thermal rating (Rt) and air-fuel ratio (Rac). These variables were calculated as follows:

$$Rt = \frac{Mw \times 10^6}{Ffuel \rho_c J F_{c1} F_{c2}} \tag{1}$$

$$Rac = \frac{Fair}{Ffuel} \tag{2}$$

where Mw represents the generated power in watts, $Ffuel$ is the total fuel flow to the boiler, ρ_c is the fuel density, J is the combustion heat, and F_{ci} and F_{c2} are conversion constants. $Fair$ is the total flow of air to the combustion. Even if more elaborated formula can be defined for these calculated variables, only the variables of table 1 could be used.

Data from 32 variables every 5 seconds were solicited to the plan personnel from several days. One day before a change of fuel, the day of change and the day after the change. However, the first problem was to deal with this huge amount of data. There are more than 17,000 registers per day. The next section describes the data pre-processing carried out.

4 Data Pre-processing

Data come from sensors and they may function improperly for a few moments o permanently. Thus, this amount of data has to be revised and cleaning. This process is referred as detecting outliers, i.e., data that is out of the expected value of this variable [3]. It is assumed that the values of each variable follow a normal distribution. The 3σ *edit rule* detection procedure is applied. This consists in calculating a distance d_i of each sample from the estimated mean with the following equation:

$$d_i = \frac{x_i - mean_i}{\sigma_x} \tag{3}$$

After cleaning the data set, a normalization is required. This is because we need to compare only the behavior between all the signals and specially between all the signals and the measured viscosity. The $min - max$ normalization was utilized, and it is given by the following formulae:

$$x' = \frac{x - min_x}{max_x - min_x}(max_{x'} - min_{x'}) + min_{x'} \tag{4}$$

where x is the unscaled variable, x' is the scaled variable, min_x is the minimum value of the unscaled variable; max_x is the maximum value of the unscaled variable; $min_{x'}$ is the minimum value of the scaled variable; $min_{x'}$ is the maximum value of the scaled variable.

Finally, a discretization is required since the probabilistic model utilizes Bayesian networks with discrete signals. After normalization, this is easy because it only needs the number of intervals in which every variable will be discretized. For example, if 20 intervals are used, then each interval consists in 0.05 of the normalized value. The next section describes the learning procedure followed after this data pre-processing.

5 Learning Probabilistic Models

There exist several learning algorithms that construct the structure of the network, and calculate the numerical parameters. The selection of the correct algorithm depends on several criteria. For example, it is required the participation of human experts in the definition of the probabilistic dependencies. Also, it is required the construction of models with relatively low interconnection. This is because the virtual sensor works on-line, i.e., fast enough.

The first problem in the learning process for the viscosity virtual sensor is the selection of the signals that may be related to the viscosity. In a first approximation, 32 variables were selected and 2 more were calculated. From these signals, an historical file with the signals every 5 seconds was obtained. This means more than 17,000 samples from 34 variables. However, this resulted in an impractical amount of information. A selection of attributes was required. This selection process was carried out with experts advice and attribute selection algorithms from weka package [4]. Table 1 describes the final set of variables, their identification and their description.

Besides the amount of columns in the data set, a lower number of records have to be selected. As may be clearly deduced, there exist long periods of time where the viscosity has no change. Only a small set of data from this period is enough.

Other special case in the learning process that has to be considered by human expert is the non trivial relation between some variables and the viscosity. For example, the thermal rating is considered but the relation of this variable with the viscosity is delayed. This means that if deficient viscosity is presented in the fuel, the result in the combustion will turn on deficient after some seconds.

Table 1. Set of variables selected to construct the model

ID	Description
T462	Internal boiler temperature
A1000	Fuel viscosity
F592X	Fuel total flow to burners
F602	Fuel return flow to tank
P600	Fuel pressure after the fuel valve
P635	Atomization steal pressure
T590	Fuel temperature
A470Y	Oxygen in the combustion gases
T4463	Atmosspheric temperature
W01	Power generated
F457	Air total flow to combustion
Z635	Aperture of the atomization steam valve
g450	Relation of additive g450 added to the fuel
eco21	Relation of additive eco21 added to the fuel
Rt	Thermal rating
Rac	Fuel - air ratio

In the current stage of the project, data from 4 days in October 2009 were obtained, pre-processed, selected and finally generate the final data set. K2 learning algorithm [1] was used for its capability of interact with human experts and because it restricts the maximum number of parents that a node may have. Figure 2 shows the resulted probabilistic model.

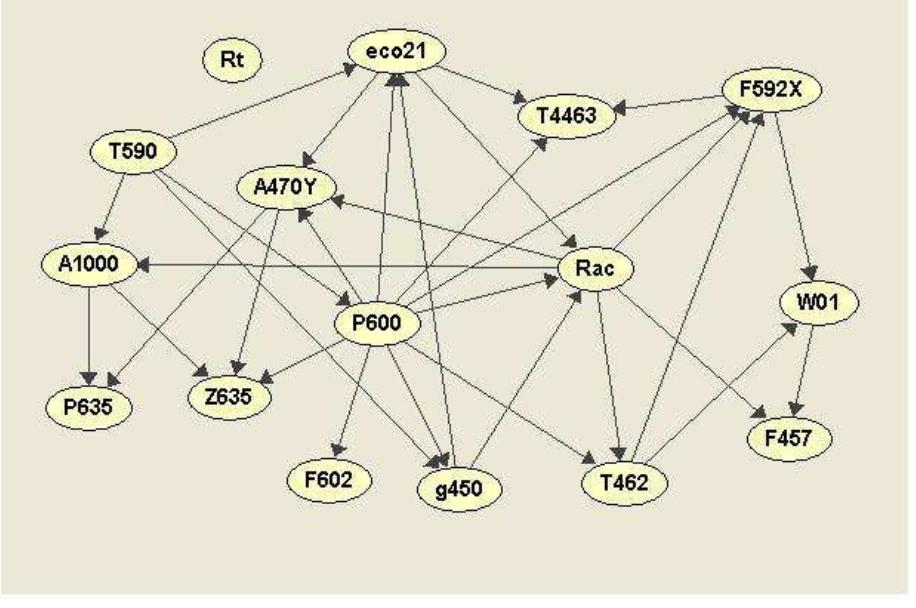


Fig. 2. Probabilistic model of the viscosity virtual sensor

This structure was discussed with human experts and approved to make the first tests.

6 Analytic Model

The viscosity of a fluid is a measure of its resistance to deformation. Most of fluids have small variations on its viscosity with temperature variations. However, fuel oil's viscosity are severe affected with the temperature variation, and due to the viscosity effect in the oil drop size required for a complete oil combustion, the fuel oil viscosity must be measured and controlled. There are so many methods to measure viscosity, however none of them guaranties a long period with reliable measurement therefore an analytic model viscosity evaluation is proposed here to complete the viscosity virtual sensor evaluation. The analytical solution comes from the energy balance, applied to a piece of tube, called Bernoulli equation:

$$\Delta P = \frac{2V^2\rho}{g_c}\left(\frac{f_L}{D} + 4k_i\right) \tag{5}$$

Fig. 3. Fuel oil viscosity lay out and outlet temperature for the viscosity control

Where the pressure drop across the piece of tube, ΔP, is related to the fluid speed, V, the tube length, L, the inner tube diameter, D and the fuel oil density ρ. g_c is the Newtons-law conversion factor, $4K_i$ is associated with the loss friction due to the tube elbows in the piece of tube. Finally, the fanning friction factor f is evaluated according to the type of fluid pattern:

$$f_L = \frac{16}{R'_e} \tag{6}$$

$$f_T = a + \frac{b}{R_e^{0.32}} \tag{7}$$

$$\frac{1}{\sqrt{f_{tr}}} + 4log_{10}\left(\frac{1}{3.7}\frac{\epsilon}{D} + \frac{1.256}{R_e}\right) = 0 \tag{8}$$

Where ϵ/D represents the relative roughness of the pipe and the subscripts L means laminar, T is for turbulent and T_r is for the transition zone flow. The fanning equation for the transition zone was taken from [7]. The viscosity-temperature relation may be represented with a semi-logarithmic equation as follows:

$$ln\mu_1 = x + \frac{B}{T_1} \tag{9}$$

Where A and B represent the constants of the straight in the semilogarithmic graph of Fig. 3.

The pressure drop measurement is done in the main heater input and using a viscosity-temperature set of graphs fuel oils, we may interpolate the actual viscosity-temperature point (μ calculated) and lay out a line, the resulting line may be extended to evaluate the temperature for the controller setpoint at the outlet main heater conditions (μobj) as is shown in Fig. 3.

7 Experiments and Discussion

The definite virtual sensor is a computer program designed to read the variables values in real time, apply them to the models and estimate the viscosity value. The first prototype of the system was written in C#. The main modules of this software are the following: (i) OPC client, (ii) integration of models, (iii) probabilistic inference, (iv) analytic inference, (v) data fusion, and (vi) user interface.

The first experiments with the viscosity virtual sensor were carried out with data sets obtained from the power plant. Using an OPC server, a plant emulator was constructed that reads a real data set from a Excel file and transmits one record at the time to the virtual sensor. The virtual sensor software never notes if it is the real central or the plant emulator that send real data.

The initial experiments include the utilization of the probabilistic model to estimate the viscosity. Evidence is loaded using data from the OPC client and the Bayesian network of Fig. 2. Probabilistic propagation is executed to produce the posterior probability distribution of the viscosity given related information. Also, real time value of the viscosity is obtained from the OPC in order to compare with the estimated and an error calculated. Remember that in this central, a hardware viscosity is installed and stored in the central data base.

Tuxpan personnel provide four days of information every five seconds. This means more than 17,000 records per day, totalizing 69,120 records in total. After the cleaning of data, 62,543 records remain. 70% of these records was utilized to train the probabilistic model, and 30% to run the tests.

Table 2 show one complete instance in the execution of the system. First, the real value of all variables is shown. All variables were discretized using ten intervals. Second, the posterior probability distribution of the viscosity is shown. The interval name is composed of the numeric interval. For example, the first interval goes from 0.073798951909 to 0.1654169701. Notice that these values correspond to the normalization of the original values. The number at the right corresponds to the probability that the viscosity value fits in this interval. Finally, the error is calculated as the percentage of the difference between the real value minus the estimated value. The estimated value considered is the half value of the interval with the highest probability. In the example of Table 2, the estimated value is 0.75503787925 normalized, that corresponds to 44.91625. In this instance, the error is 3.59%. Figure 4 show the graph of the estimated viscosity and the read viscosity and the corresponding error. For clarity, only a section with 100 estimations are shown. The scale at the right is the percentage of error. The scale at the left is the viscosity. Notice that the estimated value follows the real value with some error. In this initial set of experiments, an average of 6.2% of error was obtained (in the 18,000 cases). Several experiments will follow to increment the efficiency of the virtual sensor, i.e, minimizing the error. Some actions that may decrement the error are: (i) modify the number of intervals in the discretization, (ii) change the specifications to the K2 learning algorithm that may produce a different probabilistic model, (iii) include the contribution of the analytic model and the fusion of both estimations.

Table 2. Results of one instance of the experiments

T462 = 944.0815 F592 = 45.582 P635 = 14.9425 A470Y = 2.9711 F457 = 725.2399 g450 = 4305.2838 Rac = 15.9107	A1000 = 46.5833 P600 = 16.9673 T590 = 102.875 W01 = 185.05 Z635 = 62.79 eco21 = 8365.019
0.073798951909 - 0.1654169701	0%
0.1654169701 - 0.2561278792	0%
0.2561278792 - 0.3468387883	0%
0.3468387883 - 0.4375496974	0%
0.4375496974 - 0.5282606065	0%
0.5282606065 - 0.6189715156	0%
0.6189715156 - 0.7096824247	0.001%
0.7096824247 - 0.8003933338	60.721%
0.8003933338 - 0.8911042429	39.273%
0.8911042429 - 0.982722261091	0.003%
Visc estimated = 44.91625 Error: 3.579%	Visc read = 46.5833

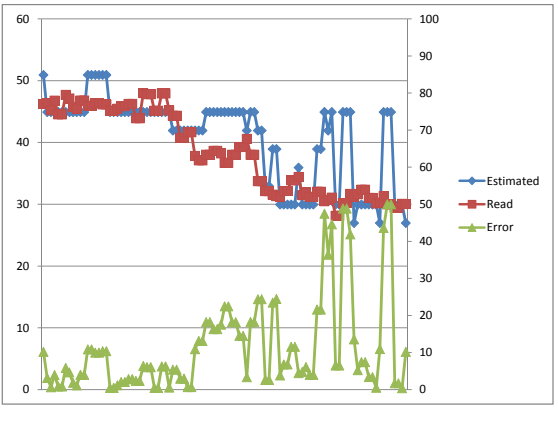


Fig. 4. Estimated viscosity, real viscosity and the corresponding error

8 Conclusions and Future Work

This paper has presented the research project for the design of an on-line viscosity virtual sensor. The system block diagram was shown and established where the main challenges are in this virtual sensor. All these challenges were explained in the sections of this paper. The experiments shown an empirical validation of the system using different data sets also obtained from the power plant. The results are promising. The definite installation of the virtual sensor is planned by

the end of the year 2010. Future work will be done when validating the virtual sensor in the plant and some adjustments can be necessary. The most important future work will be the design of an automatic temperature control that guarantees that the fuel feed to the boiler is at the optimal viscosity in order to obtain an optimal combustion.

Acknowledgments

This research work is supported by the grant 89104 from CONACYT-CFE Sectorial Found.

References

1. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning 9(4), 309–348 (1992)
2. Comision Federal de Electricidad, Generación termoeléctrica (2008), `http://www.cfe.gob.mx/es/LaEmpresa/generacionelectricidad/termoelectrica`
3. Fortuna, L., Graziani, S., Rizzo, A., Xibilia, M.G.: Soft Sensors for Monitoring and Control of Industrial Processes. In: Advances in Industrial Control. Springer, Glasgow (2007)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explorations 11(1) (2009)
5. Ibargüengoytia, P.H., Reyes, A., Huerta, M., Hermosillo, J.: Probabilistic virtual sensor for on–line viscosity estimation. In: Gelbukh, A., Morales, E. (eds.) Seventh Mexican International Conference on Artificial Intelligence, Piscataway, N.J., U.S.A., pp. 377–382. IEEE Computer Society, Los Alamitos (2000)
6. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco (1988)
7. Thomas, P.: Simulation of Industrial Processes for Control Enginners. Butterworth Heinemann, Butterworths (1999)

Evaluating an Affective Student Model for Intelligent Learning Environments

Yasmín Hernández[1], Luis 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 model for intelligent tutoring systems; thus, the tutor considers the affective state as well as the knowledge state of the student to give instruction to students. An important component of the affective model is the affective student model. This last one is rooted on the OCC cognitive model of emotions and the five-factor model, and it is represented as a dynamic Bayesian network. The personality traits, goals and knowledge state are considered to establish the student affect. The affective model has been integrated to an intelligent learning environment for learning mobile robotics. We conducted an initial evaluation of the affective student model with a group of 20 under graduate and graduate students to evaluate the affective student model. Results are encouraging since they show a high agreement between the affective state established by the affective student model and the affective state reported by the students.

Keywords: intelligent tutoring systems, student's affect, student modeling.

1 Introduction

Emotional states have been recognized as important components in motivation and learning. There is evidence that experienced human tutors monitor and react to the emotional state of the students in order to motivate them and to improve their learning process [1]. In recent times, there has been extensive work on modeling student emotions in intelligent tutoring systems, see [2, 3, 4]; however, there have been only limited attempts to integrate information on student affect in the tutorial decisions, e.g. [5, 6, 7]. If we want to offer students the best tutorial action given both the students' affective and knowledge state: firstly, we have to identify the student affective state and then to establish the suitable tutorial action.

To tackle this problem, we have developed an affective behavior model (ABM) that considers both the affective and knowledge state of the student to generate tutorial actions. The ABM selects the tutorial actions with the best expected effect on a student's affect and knowledge by using a dynamic decision network with a utility

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 473–482, 2010.

measure on both, learning and affect. We have designed the ABM based on interviews with qualified teachers aimed at understanding which actions the teachers select according to the state of a student's affect and knowledge.

An important component of the ABM is the *affective student model*; it is based on a probabilistic affective model previously developed by researchers at the University of British Columbia, [2, 8, 9]. The model is based on the OCC model of emotions, the Five-Factor model of personality, and intuition. Previously to assess the complete ABM, we wanted to prove the affective student model, we conducted a pilot study to evaluate the affective student model, we compare the affective student state establishing by the model and the affective state reported by the student. An initial approach and evaluation of the model is presented on [10]. Conversely, we have also implemented the affect student model in an educational game to learn numbers factorization [2, 8, 9], a description of this case of study is presented in [11]. In this domain the students interact with the environment by playing, and they receive instruction with base on the affective student model and on the pedagogical student model. We also evaluated the model and we obtained encouraging results [11].

In the literature we have find similar works; for example, in [12] it is established a relationship between facial expression and affective states, they conducted a study videotaping students and asking them for their affective state. An approach based on physical sensors is presented in [13]; they establish the affective state of students considering the signals from a pressure mouse, a posture analysis seat, a skin conductor sensor and a facial expression camera. A similar work to our study to test the affective model is presented in [l4]; they carried out a survey comparing the self judgments of students versus teachers' judgments. All these works, this one included, share a common aim: to establish a relationship between the affective state and the available information about the individuals. However, they use different methods, the main different is that approaches rely on the use of physical signals or judgments of a third party; while we use contextual information and the personality traits of the individual. An advantage of our methodology is that students do not have to use special equipment to interact with the systems; when students have to use any equipment besides the computer they can feel observed, so possibly they are not authentic in their behavior and errors in the recognized state can occur.

In this paper, we describe the construction and the evaluation of the affective student model. The rest of the paper is organized as following: section 2 briefly describes the ABM; in section 3 we present the affective student model; in section 4 we describe how the affective model is constructed; section 5 describes the pilot study, and finally we present our conclusions.

2 Modeling Affective Behavior

The affective behavior model we have developed for intelligent tutoring systems takes affect into account when interacting with a student by i) inferring the affective state of the student (affective student model); and ii) by establishing the optimal tutorial action based on the student's current affective and knowledge state (affective tutor model). A flow diagram of the ABM is presented in Fig. 1.

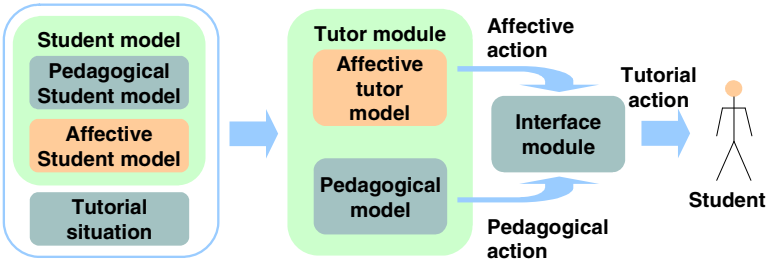


Fig. 1. General diagram for the Affective Behavior Model. The model is composed by an affective student model and an affective tutor model. The tutor model produces an affective action considering the affective and pedagogical student models and the tutorial situation. The affective action is a component of the tutorial action to be presented to the student.

To generate a tutorial action, the ABM considers the affective and knowledge state of the student (as assessed by the corresponding student models), as well as the tutorial situation. The tutorial action is viewed as consisting of two components. The first component targets mainly the student's affective state (affective component) and consist of an agent's animations. The second component (pedagogical component) targets mainly the student's knowledge and consists of textual lessons. Thus selecting a tutorial action involves selecting these two components. Finally, the interface module establishes the physical realization of the tutorial action. To assess the affective state of the student, the ABM relies on a probabilistic student model that bases its predictions on information about a student's personality and interaction behavior. In the next section we describe this model.

3 Affective Student Model

As we mentioned, one of the first steps to have affective behavior in an intelligent tutor is to know the affect of the student; therefore we developed an affective student model with base on the OCC model (briefly described below) and on an affective student model previously defined by [2, 8, 9].

Our affective student model uses the OCC model [15] to provide a causal assessment of student's emotions based on contextual information. The OCC model defines emotional state as the outcome of the cognitive appraisal of the current situation with respect to one's goals, principles and preferences. Emotions of the students are for the elements in the current situation: events, agents and objects and for herself. Fig. 2 aims to show the basics of the OCC model.

The affective student model consists of a dynamic Bayesian network (DBN) that probabilistically relates student personality, goals and interaction events with student's affective states, based on the theory defined by the OCC model. Fig. 3 shows a high level representation of the model, where each node in the network is actually a set of nodes in the actual 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 in each individual. The dynamic Bayesian network models this dynamic nature of the affective state and its

influence in the next state. The network includes two time slices at any given time. A slice is added and a slice is discarded after each student's action. To infer the affective state at t_n we use the knowledge state of the student, the tutorial situation and the personality traits of the student; this is used to predict the affect at t_{n+1}.

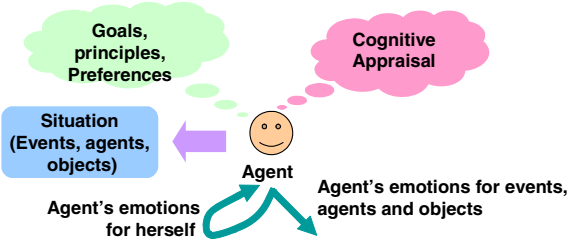


Fig. 2. Simplified diagram of the OCC Cognitive Model of Emotion [10]. The emotions are for the elements included in the actual situation, they can be events, other agents and objects.

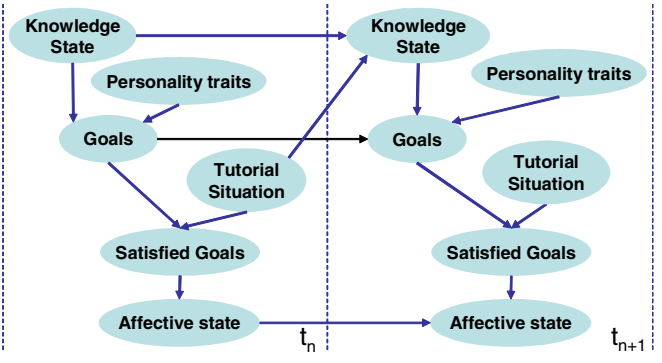


Fig. 3. High level DBN for the affective student model. We include two time slice to represent the dynamic behavior of affect and its impact in the next state.

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. Next, we describe the DBN, how we build the model and how we obtain the nodes values.

4 Building the Affective Model

Following we describe how we build the student model in a test case. To evaluate the affective student model we used an intelligent learning environment to learn mobile robotics. In this environment the students learn by carry out experiments about to set up and to guide an mobile robot to reach the goal; and after to complete the experiments they have a lesson according their performance [16].

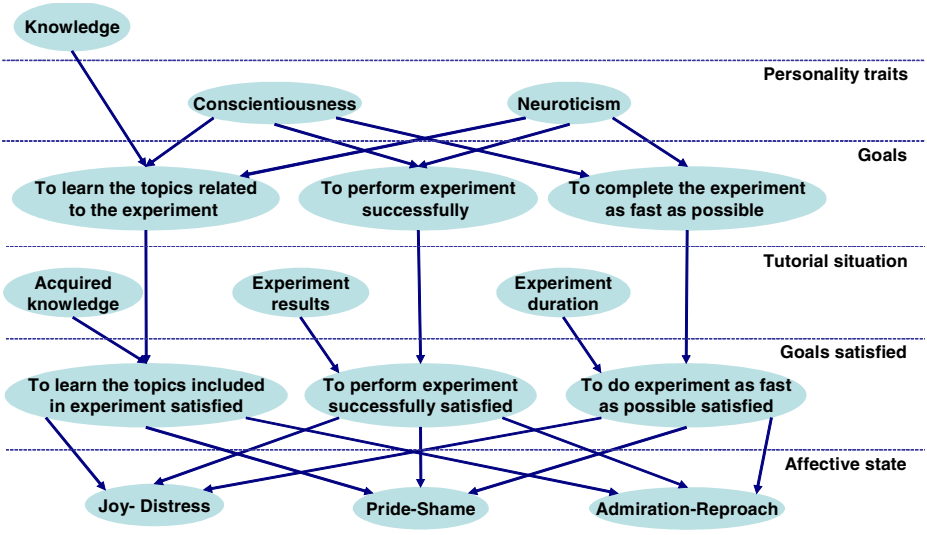


Fig. 4. Detailed affective student model represented by a Bayesian network. Each set of nodes is a detailed representation of the DBN at an specific time.

As we mentioned above, each node in the DBN in Fig. 3 is actually a set of nodes; Fig. 4 shows the detailed Bayesian network for the affective student model in the study case for mobile robotics. The dependency relations in the Bayesian network have been established based on the literature [17, 18, 19], teachers' expertise and intuition, following we describe them.

According to the OCC model, the goals are fundamental to determine the affective state; however, we think they cannot be explicitly asked to the student during the interaction, because in order for the student to provide a reliable answer she would need to understand the question and do self-inspection, and errors can occur. Consequently, the goals in our model are inferred from indirect sources of evidence, we use personality traits and student's knowledge as a predictor of a student's goals.

We based the personality traits on the Five-Factor Model [18, 19], which considers five dimensions for personality: openness, conscientiousness, extraversion, agreeableness and neuroticism. From the five dimensions for personality, we include only two, *conscientiousness* and *neuroticism*, to establish goals. We included only these dimensions because to establish goals because they are the ones for which a stronger relationship with learning [19]. Also, a relationship between *openness* and learning has been reported, but it has not been proven [19].

To obtain priors on the personality nodes in the network, we conducted a study with 58 undergraduate and graduate students. This group of students is a representative sample of the population who is intended for using the learning environment: they are graduate or undergraduate students, they are attending the same subjects and they are in the same age group. In this study, the students had to answer a personality test based on the five-factor model [20]. The test consists of 50 adjectives, each adjective is an item of the test and the individuals have to rate how much the adjective can be applied to them. The grade of the test indicates if the student is in

Table 1. Results of the personality study for the dimensions *conscientiousness* and *neuroticism*

Values	Conscientiousness	Neuroticism
High	0	0
Average	78	78
Low	22	22

a low, average o high level for each personality dimension. The results of the study are presented in Table 1. As we can see the most of the students (78%) are in average level and a smaller student group (22%) is in low level for both personality dimensions. In this sample no one student is in high level for no one personality dimension (we can observe that both personality traits have the same percentages).

Nevertheless, besides to have these priors, to have a more precise estimation of personality, we had the participating students answering the same personality test before to use the system.

The dependency relations in the DBN (Figure 4) have been established based on the literature [17, 18, 19] and on insights from teachers. For example, if the student has a conscientiousness personality and limited understanding of number factorization, the probability of having the goal to learn the topics related to the experiment high, because she is a responsible person who cares for her duties. On the other hand, if the student is a neurotic person, there is a higher probability of having the goal to succeed in reach the goal than to learn, because a neurotic person wants to have immediate and tangible success.

To infer the students' goals we also include the knowledge the student have about the present topic in the current experiment, besides the student's personality.

With base in the previous description: In the affective model we included three goals: 1) *to learn the topics related to the experiment*, 2) *to perform experiment successfully*, and 3) *to complete the experiment as fast as possible*. The reasons for establishing these goals are based on the nature of the task: to perform an experiment to learn mobile robotics. The first goal can be present because it is the main aim of the task: to complete an experiment for learning; the second goal can be present because the student can want to have success in reaching a goal; and the third goal can be present because generally students want a quick reward.

The information for the knowledge state and tutorial situation nodes comes from the student action results by means of the pedagogical student model.

For the tutorial situation nodes, we use the knowledge about the topics included in the experiment and data such as how much it takes to get the goal, how many times the student make a correction in her way, as they are presented in Fig. 5.

With base on the OCC Model, 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* nodes. The influence of the appraisal process on the student's affect is represented by the link between the *satisfied goals* nodes and the *affective state* nodes. From the complete set of emotions proposed by the OCC model, the affective student model only includes six emotions: *joy*, *distress*, *pride*, *shame, admiration* and *reproach*. These are represented as three pairs of mutually exclusive emotions (for the same event/situation): *joy-distress*, *pride-shame* and *admiration-reproach*, each represented by a binary node in the network.

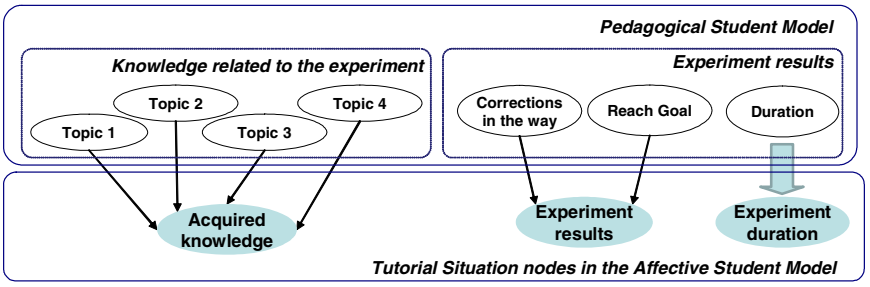


Fig. 5. Tutorial situation nodes. The tutorial situation nodes consider the pedagogical student model and the experiments results.

The *joy-distress* affective state node represents the emotions the student can have for the situation: she is happy because she learnt, or because she got the goal or because she completed the experiment quickly.

The *pride-shame* affective state node represents the emotions from student to herself: she is pride because she learnt the topics in the experiment, or because she completed the experiment successfully or because she got the goal quickly.

The *admiration-reproach* affective state node represents the emotion from the student to the tutor represented as an animated agent. As a part of our study we included in the learning environment an animated agent to present the instruction and to be the face of the tutor. The student can feel admiration for the tutor because the tutor taught her, and therefore she got the goals. Fig. 6 describes how the emotions arise in the affective student model according the OCC Model. The (student or tutor) agent take out an action and the student see the result, she compares the results with her goal, and the emotions will arise according with how much the situation fits with the student's goal.

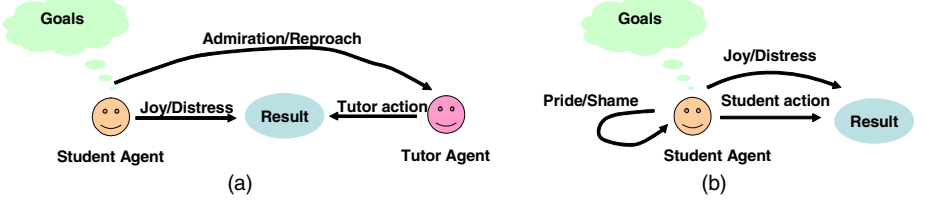


Fig. 6. Emotion occurrences in the affective student model according to the OCC model. In (a) it is shown the process for emotion dimensions; each pairs are mutually exclusive mutually emotions joy-distress, pride-shame and admiration-reproach.

5 Case Study

To evaluate the performance of the affective student model, we have conducted a Wizard of Oz study with a group of students. The aim of a Wizard of Oz study is to obtain information for designing and evaluating prototypes or systems which have not finished yet [21]. This type of study have been used since human-computer

interaction began and they have used widely for emulate technologies in interactive systems [21, 22, 23]. They consist on to use operators or mechanisms hidden from the user for temporally to emulate not finished yet components of a computer system during its development [21].

In our study we had 20 undergraduate and graduate students participating. This is a small but representative sample of the type of students who will be using the learning environment. At the same time, to evaluate the study results we use the *t test* which is intended for small samples, even less than 20 students [24].

As first point in the survey, the students answered a personality test, based on the Five-Factor model [20]. The personality test is composed by 50 adjectives. The students have to rate how each adjective applies to them. The survey consisted in presenting to students three different tutorial scenarios. The tutorial scenario includes the development of an experiment and a tutorial action presented by an animated character. The tutorial action was selected considering the affective student state and the tutorial situation presented in the scenario. After presenting each tutorial scenario, the students were asked about their affective state given the tutorial situation with the aim to compare with the affective state established by the affective student model. The questions presented to the students to ask their affective state is shown in Table 2.

We compared the affective state reported by the students with the affective state established by the affective student model; the results are summarized in table 3. We found for the emotion *joy-distress*, the model established the affective state correctly in a 72% of the cases; for the emotion *pride-shame*, the model established the affective state correctly in a 70% of the cases; and for the emotion *admiration-reproach*, the model established the affective state correctly in a 55% of the cases.

Table 2. Question presented to the students. The students were asked to answer this question after each tutorial session, with the aim to compare with the affective student model with their reports.

Please, rate your affective state after to complete the tutorial session:		
- Towards the learning environment:		
Joy	5----------4---------3--------2---------1	Distress
- Towards the tutor:		
Admiration	5----------4---------3--------2---------1	Reproach
- Towards yourself		
Pride	5----------4---------3--------2---------1	Shame

Table 3. Percentage of agreement between the affective state established by the affective student model and the affective state reported by the students

	Joy-distress	**Admiration-reproach**	**Pride-shame**
Agreement	43	33	42
Disagreement	17	27	18
Percentage of agreement	71.67%	55%	70%

The emotion *joy-distress* is the emotion of the student towards the activity; the emotion *pride-shame* is the emotion of the student towards herself; and the emotion *admiration-reproach* is the emotion of the student towards the tutor. As we can see,

the model was precise for the emotions *joy-distress* and *pride-shame*; however, for the emotion *admiration-reproach*, the model was not precise. We suppose that is because the emotions from students towards teachers emerge slowly. We suppose that students think that they are learning and, in general, they do not think that teachers are instructing them well. This is coincident with comments of teachers, in other survey, who rated the emotion from students to them mostly negative. Another reason can be that students had not enough information to evaluate this emotion because they did not receive a quiz to evaluate their knowledge. We have to conduct further investigations to validate this hypothesis and refine the ABM. We are considering including a test in the next user study. These results are encouraging; however, these results are not conclusive as they are based on a simulation of the ABM in a Wizard of Oz experiment. The next stage in our research is to integrate the ABM with the tutor, and conduct a controlled user study.

6 Conclusions 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 affective student model 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. 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.

Acknowledgments. We would like to thank Cristina Conati from the University of Bristish Columbia, for many useful discussions on the definition of the affective model. This research was supported in part by the *Instituto de Investigaciones Eléctricas*, Mexico.

References

1. Johnson, W.L., Rickel, J.W., Lester, J.C.: Animated Pedagogical Agents: Face-to-Face Interaction. Interactive Learning Environment, International Journal of Artificial Intelligence in Education 11, 47–78 (2000)
2. Conati, C., Mclaren, H.: Empirically Building and Evaluating a Probabilistic Model of User Affect. User Modeling and User-Adapted Interaction 19, 267–303 (2009)
3. 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)
4. Alexander, S., Hill, S., Sarrafzadeh, A.: How do Human Tutors Adapt to Affective State? In: UM 2005, Workshop on Adapting the Interaction Style to Affective Factors (2005)
5. Zakharov, K., Mitrovic, A., Johnston, L.: Towards Emotionally-Intelligent Pedagogical Agents. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 19–28. Springer, Heidelberg (2008)
6. Faivre, J., Nkambou, R., Frasson, C.: Toward Empathetic Agents in Tutoring Systems. In: Russell, I., Haller, S. (eds.) FLAIRS 2003, pp. 161–165. AAAI Press, Florida (2003)
7. Murray, R.C., VanLehn, K.: DT Tutor: A Decision-theoretic, Dynamic Approach for Optimal Selection of Tutorial actions. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) ITS 2000. LNCS, vol. 1839, pp. 153–162. Springer, Heidelberg (2000)

8. Conati, C., Zhou, X.: Modeling students' emotions from Cognitive Appraisal in Educational Games. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 944–954. Springer, Heidelberg (2002)
9. Conati, C., Mclaren, H.: Data-driven Refinement of a Probabilistic Model of User Affect. In: Ardissono, L., Brna, P., Mitrovic, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 40–49. Springer, Heidelberg (2005)
10. Hernández, Y., Noguez, J., Sucar, L.E., Arroyo-Figueroa, G.: A Probabilistic Model of Affective Behavior for Intelligent Tutoring Systems. In: Gelbukh, A., de Albornoz, A., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 1175–1184. Springer, Heidelberg (2005)
11. Hernández, Y., Sucar, L.E., Conati, C.: Incorporating an Affective Behavior Model into an Educational Game. In: Lane, H.C., Guesgen, H.W. (eds.) FLAIRS 2009, pp. 448–453. AAAI Press, Florida (2009)
12. Abbasi, A.R., Uno, T., Dailey, M.N., Afzulpurkar, N.V.: Towards Knowledge-based Affective Interaction: Situational Interpretation of Affect. In: Paiva, A., Prada, R., Picard, R.W. (eds.) ACII 2007. LNCS, vol. 4738, pp. 452–463. Springer, Heidelberg (2007)
13. Dragon, T., Arroyo, I., Woolf, B.P., Burleson, W., el Kaliouby, R., Eydgahi, H.: Viewing Student Affect and Learning through Classroom Observation and Physical Sensors. In: Woolf, B.P., Aimeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 29–39. Springer, Heidelberg (2008)
14. D'Mello, S., Taylor, R., Davidson, K., Graesser, A.: Self Versus Teacher Judgments of Learner Emotions During a Tutoring Session with AutoTutor. In: Woolf, B.P., Aimeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 9–18. Springer, Heidelberg (2008)
15. Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press, Cambridge (1988)
16. Noguez, J., Sucar, E.: A Semi-open Learning Environment for Virtual laboratories. In: Gelbukh, A., Albornoz, A., Terashima, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 1185–1194. Springer, Heidelberg (2005)
17. Boeree, G.: Personality Theories, Electronic Textbook, http://www.ship.edu/~cgboeree
18. Costa, P.T., McCrae, R.R.: Four Ways Five Factors are Basic. Personality and Individual Differences 13(1), 653–665 (1992)
19. Heinström, J.: The impact of personality and approaches to learning on information behaviour. Electronic Journal, Information Research 5(3) (2000), http://informationr.net/ir
20. Boeree, G.: Big Five Minitest, On-line test (1998), http://www.ship.edu/~cgboeree/
21. Dow, S., MacIntyre, B.: New Media Collaboration through Wizard-of Oz Simulations. In: CHI 2007, Workshop: HCI and New Media Arts: Methodology and Evaluation (2007)
22. Dow, S., MacIntyre, B., Lee, J., Oezbek, C., Bolter, J.D., Gandy, M.: Wizard of Oz Support throughout an Iterative Design Process. IEEE Pervasive Computing, Special Issue on Rapid Prototyping 4(1), 18–26 (2005)
23. Anderson, G., Hook, K., Mourao, D., Paiva, A., Costa, M.: Using a Wizard of Oz study to inform the design of SenToy. In: Mackay, A., Amowitz, J., Gaver, W., Verplank, B., Sutcliffe, A. (eds.) DIS 2002, pp. 349–355. ACM, New York (2002)
24. Navidi, W.: Estadística para ingenieros y científicos. McGraw-Hill Interamericana, México (2006)

Dynamic Reward Shaping: Training a Robot by Voice

Ana C. Tenorio-Gonzalez*, Eduardo F. Morales, and Luis Villaseñor-Pineda

National Institute of Astrophysics, Optics and Electronics,
Computer Science Department,
Luis Enrique Erro #1, 72840 Tonantzintla, México
{catanace17,emorales,villasen}@inaoep.mx
http://www.inaoep.mx

Abstract. Reinforcement Learning is commonly used for learning tasks in robotics, however, traditional algorithms can take very long training times. Reward shaping has been recently used to provide domain knowledge with extra rewards to converge faster. The reward shaping functions are normally defined in advance by the user and are static. This paper introduces a dynamic reward shaping approach, in which these extra rewards are not consistently given, can vary with time and may sometimes be contrary to what is needed for achieving a goal. In the experiments, a user provides verbal feedback while a robot is performing a task which is translated into additional rewards. It is shown that we can still guarantee convergence as long as most of the shaping rewards given per state are consistent with the goals and that even with fairly noisy interaction the system can still produce faster convergence times than traditional reinforcement learning techniques.

1 Introduction

Service robots are beginning to be used in a wide range of applications such as entertainment, assistance, maintenance, cleanse, transport, and guidance, and it is expected to be commonly found in houses and offices in the near future. A key element for their complete incorporation will be their ability to learn new tasks. Reinforcement Learning (RL) [21] has been widely used and suggested as a good candidate for learning tasks in robotics, e.g., [2,20,16,23]. This is mainly because it allows an agent, i.e., the robot, to "autonomously" develop a control policy for performing a new task while interacting with its environment.

The use of reinforcement learning in robotics can be problematic due to large state and action spaces which normally results in long training times. Different approaches have been suggested to produce faster convergence times, such as the use of abstractions, hierarchies, function approximation, and more recently reward shaping.

An alternative approach to teach a robot how to perform a task is with programming by demonstration, in which the user shows a robot an action which is

* Master's thesis scholarship No. 214262 and research project No. 84162 from Conacyt.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 483–492, 2010.

later executed by the robot. This approach, however, normally uses sophisticated hardware and can only reproduce the traces provided by the user.

An alternative, and we believe more natural, approach to make a robot learn a new task, is by telling the robot how to perform it and provide feedback while the robot is executing the task until we are satisfied with its performance. In this paper, the user's feedback is associated with a reward shaping function and is used to produce faster convergence times. This approach, however, pose several problems:

1. Even with simple voice commands, speech recognition systems are not completely reliable which can produce a noisy reward shaping function, so we would like to know under which conditions it is possible to recover from such errors.
2. The user can provide his rewards with certain delay, so we would like to know how the delay affects our system.
3. The user is not consistent with his feedback and it can vary over time, at first providing a substantial amount of feedback and after a while just giving occasional suggestions, so we would like to know if it is important to take into account such behavior in our system.
4. Under the above mentioned conditions, can we still guarantee convergence and if so, under which conditions?

This paper addresses most of these issues. The rest of the paper is organized as follows. Section 2 describes related work. In Section 3 we provide the theory required to answer the previous questions. Section 4 describes the experimental results with a mobile robot performing navigation tasks and Section 5 concludes and suggests future research directions.

2 Related Work

There is a large amount of literature describing reinforcement learning techniques in robotics. In this section we only review some of the most closely related work to our proposal. Reinforcement learning including feedback has been considered in some approaches [23,11,6,10,12]. Hardware devices such as joysticks, keyboards, among others are used to provide feedback. Other closely related approaches use voice commands to teach how to perform a task, as presented in [19,24]. In [19] a human instructor demonstrates how to do a task and gives instruction by speech. But, verbal instructions are very similar to control structures of a programming language that can be difficult to give by common people, and the learning is focused on learning by demonstration. By contrast, the method that we propose uses a more natural spoken interaction and uses reinforcement learning in addition of demonstration. In [24] an Actor-Critic algorithm based on IHDR (Hierarchical Discriminant Regression) is presented. A teacher sets the robot's arm in different positions, each one is named with a particular word (command) and the complete sequence of positions is named too. A position or a sequence of positions can then be invoked with its identifier. A feedback is given

during the training phase, the teacher uses a bumper to mark the positions and to provide some reinforcements, so is necessary an interaction with hardware. In this paper, we propose a method without any special hardware requirement of the robot and with a more natural interaction approach. Unlike other approaches, our method uses language in a more natural way to guide the demonstration, and uses reinforcement learning with verbal feedback as dynamic reward shaping.

Reward shaping has been used to accelerate reinforcement learning [15,9]. More recently, researchers have try to learn a reward shaping function [1,7,4,3,13], with designs of structural shaping (decomposition of a complex task in simple tasks) as [5,14,7,18], and with shaping that increases the complexity of a task over time as [5,14]. However, most of these methods require domain knowledge to design an adequate reward shaping function, which are static over time, or try to learn the functions with experience, which can take very long training times. Some authors have provide traces from a user that can be used to guide the reinforcement learning process [1,8]. Our method also uses traces provided by a teacher but can also incorporated verbal feedback during the learning process.

Contrary to other approaches, our method uses initial traces and feedback (commands and qualifiers) provide by a teacher through voice that can be given at any time during the learning process. It deals with continuous spaces and can reach good policies with moderated noisy feedback. In the following sections we describe in detail the proposed method.

3 Dynamic Reward Shaping

In this section we introduce our formal framework. Reinforcement learning problems can be modelled as Markov Decision Processes (MDPs). Given an MDP, $M = (S, A, T, R)$, what we want to learn is a policy, i.e., the best action to perform on each state to receive the maximum expected accumulated reward. The idea of reward shaping is to give additional rewards to a learning agent to guide its learning process and converge faster. In effect, the learning algorithm is running on a transformed MDP, $M' = (S, A, T, R')$, where $R' = f(R, s, s')$. Normally f has been used as an additive function, i.e., $R' = R + F$, but in general, it does not need to be the case. So in the new MDP when an agent takes in state s action a and moves to s' it receives a reward defined as $R + F$.

A significant advancement in the formalization of reward shaping was the development of potential-based reward shaping [15]. Here the idea is that the reward for executing a transition between two states is mainly the difference in the value of a potential function ϕ applied to each state. It can obtain optimal policies and solve some problems reported in [18] with shaping functions. In particular, the difference of potentials avoids creating loops between a sequence of states when an agent gain positive rewards that deviates it from the goal. The function F is represented under this framework as: $F(s, s') = \gamma\phi(s') - \phi(s)$.

Potentials are adequate if they are given when the agent gets closer towards the goal and are normally defined by the user. Recently there has been some work on how to learn a potential, by learning a value function over a simplified

problem and use this value function as reward shaping or by using an estimate of the distance to the goal, similarly as the underestimate measures used in search techniques [13]. In simple mazes this can be a Manhattan distance between the agent and the goal, however, they can produce wrong estimates if obstacles are ignored, as getting closer in straight line can actually take the agent away from the goal.

3.1 On-Line Feedback by the User

In our case, we assume that the shaping rewards are given by the user through voice commands. The user can qualify the behavior of the robot, for instance, good or bad, which are then translated into rewards, or can command the robot to perform a particular action. Algorithm 1 describes the pseudo-code of SARSA learning with dynamic rewards and other algorithms could be adapted as well.

Algorithm 1. SARSA learning with dynamic rewards.

```
Initialize Q(s, a) arbitrarily
for each episode do
  Initialize s
  repeat
    for each step in episode do
      if User selects action a in s then
        Take action a, observe r, s'
        Let r ← h(r, a) {where h(r, a) is a bounded function that depends on the
        selected action}
      else
        Select an a from s derived from policy Q (e.g., ϵ–greedy)
        Perform action a, observe r, s'
      end if
      if user provides feedback f on state s' then
        Let r ← g(r, f) {where g(r, f) is a bounded function that depends on the
        provided feedback}
      end if
      Select a' from s' derived from policy Q (e.g., ϵ–greedy)
      Q(s, a) ← Q(s, a) + α [r + γQ(s', a') − Q(s, a)]
      s ← s';
    end for
  until s is a terminal state.
end for
```

When the user is providing feedback on-line, there can be several cases:

Permanent feedback: If we assume that the user is continuously providing adequate feedback to the robot, then we can use results from difference of potentials. It is expected that, at least for many domains, the user has a clear idea of what represents progress towards the goal. We only scale-up the value function and learn the same policy.

Sporadic feedback: In our case, the user may provide voice commands, and consequently affect the rewards, but not all the time and not on the same state-action pairs. In this case we have sporadic rewards. This occasional rewards, depending on their frequency, will fade-out with time and with distance. However, if we have a reward persistently given in a particular state, this will create a new sub-goal. In this way, the user can guide the robot towards intermediate goals. If we have relatively frequent rewards all over the space, we can still converge towards the optimal policy as long as the expected rewards on each state follow a potential-based reward function. In general, this is not too difficult to have as we expect the user to have a clear idea of when the robot is actually progressing towards the goal.

Noisy feedback: Now suppose the user sometimes gives a wrong reward, gives it out of time or the speech recognition system misinterprets the command. In any case the agent receives a wrong reward. These situations can deviate the agent towards a wrong policy. In this case, the agent can still recover provided the wrong rewards are not given all the time and are stopped while there are good chances of following exploratory movements. Also, the user can try to correct the errors with additional feedback. In this sense, the user can always impose his own policy. If the rewards are given in average towards the goal during the learning process, then we can still produce an optimal policy as before, although it may take longer training than a strategy without reward shaping.

In all cases, our reward function is defined as: $R' = R + F$ where F is a reward provided by the user. The main difference with previous reward shaping functions is that in our case the rewards can be given sporadically and can be contrary to what it is needed for achieving a goal. Nevertheless, we assume that when they are given correctly they reinforce the movements where the agent is moving towards the goal and satisfy a potential-based shaping framework. So even with noisy feedback from the user we can still guarantee convergence towards an adequate policy as long as the agent receives in average correct rewards.

4 Experiments

In this section, we illustrate our approach using an autonomous mobile robot that learns navigation tasks in simulated environments. We used Player/Stage to simulate an autonomous mobile robot Pioneer 2 for the experiments. The robot has a frontal laser and a sonar in the rear part. And, we used Sphinx 3 to the speech recognizer based on the corpus DIMEx100 [17].

In order to learn a new task, the user first tells the robot what actions to do in order to complete the task. The user can provide one or more initial traces that are used as guidance to the robot. With these initial traces the robot follows a reinforcement learning process to try to improve the performance over these traces as they are expected to have errors and be far from optimal. During learning the robot receives rewards by the the traditional reward function defined in reinforcement learning but can also received additional rewards from user's

feedback. In the experiments our reward function is given by: $R = R_{RL} + R_U$, where R_{RL} is the traditional reinforcement learning reward function and R_U is the reward associated with the user's voice commands. We used Algorithm 1 with eligibility traces for learning.

In our algorithm, the states are represented with information from the robot's sensors and are incrementally created while the robot is traversing the environment. When the readings of the sensors are sufficiently different from previous stored states a new state is created. We use Pearson's coefficient as similarity function and a threshold value to decide if the current sensor readings is sufficiently different from the current stored states to create a new state. Each state is associated with a set of discrete actions but our framework is able to produce continuous actions by combining discrete actions considering their current Q-values (more details of the representation and the way to produce continuous actions are given in [22]).

The language chosen to interact with the robot was Spanish. The vocabulary (with around 250 words) was composed of words which we later expanded in a second corpus to deal with short phrases. The words of interest were qualifiers and commands of actions. Part of it is showed in Table 1.

Table 1. Examples of vocabulary with a rough translation into English

WORDS	SHORT PHRASES
Avanzar (forward)	Hacia adelante (move forward)
Regresar (backward)	Hacia atrás (move backwards)
Izquierda (left)	Gira a la izquierda (turn to your left)
Derecha (right)	Ve a tu derecha (go to your right)
Fin, Final (end)	Para ahí (stop there)
Bien (good)	Sigue así (keep like this)
Mal (bad)	Por ahí no (not that way)
Excelente (excellent)	Muy bien (very good)
Terrible (terrible)	Así no (not like that)
Objetivo (goal)	Hasta ahí (until there)

In the experiments, we associated rewards to certain words of the vocabulary: +100 for reaching the goal (*objetivo*), +50 for "excellent" (*excelente*), +10 for "good" (*bien*), −50 for "terrible" (*terrible*), and −10 for "bad" (*mal*). Similar rewards were used to the rest of the vocabulary and the phrases.

We performed several robot navigation experiments. Figure 1 shows the different tasks that were taught to the robot. We compared the performance of the following settings:

1. SARSA(λ)
2. SARSA(λ) with user's feedback considering errors (around 20% noisy) of our current speech recognition system

Fig. 1. Tasks 1-4 (left to right, up to down) that were taught to the robot

3. SARSA(λ) with user's feedback and with initial guided traces considering errors (around 20% noisy) of our current speech recognition system
4. SARSA(λ) with user's feedback and with initial guided traces without interpretation errors from the speech recognition system

Each experiment was repeated three times and averages are shown in the results. Figure 2 and Table 2 show the total times of the learning process for learning the four tasks. As can be seen, the number of episodes is roughly similar in the three cases (although it is smaller with feedback and initial traces) but the learning times are much faster when the algorithm uses feedback from the user (3.84 times faster than traditional reinforcement learning when we used feedback and 5.4 times faster when we used feedback and initial traces).

Table 3 shows to the left the number of interventions (feedback) provided by the user on each task. As can be seen in average the number of interventions is reduced in half if initial traces are given beforehand. Table 3 shows to the right a comparison between learning with perfect feedback (no mistakes from the speech understanding system) and with noisy (around 20% noisy) feedback for Task 2.

Table 2. Total number of episodes (left) and computing times (right) of the different experiments per task. RL = SARSA learning, RL + F = RL with feedback from the user, and RL + T + F = RL + F with the initial traces provided by the user.

	RL	RL + F	RL + T + F
T1	13	9	6
T2	6	7	7
T3	12	12	7
T4	7	12	11
Avg	9.5	10	7.75

	RL	RL + F	RL + T + F
T1	103.93	19.59	12.85
T2	66.4	15.1	13
T3	100.65	38.2	18.09
T4	99.1	23.43	24.61
Avg	92.54	24.08	17.13

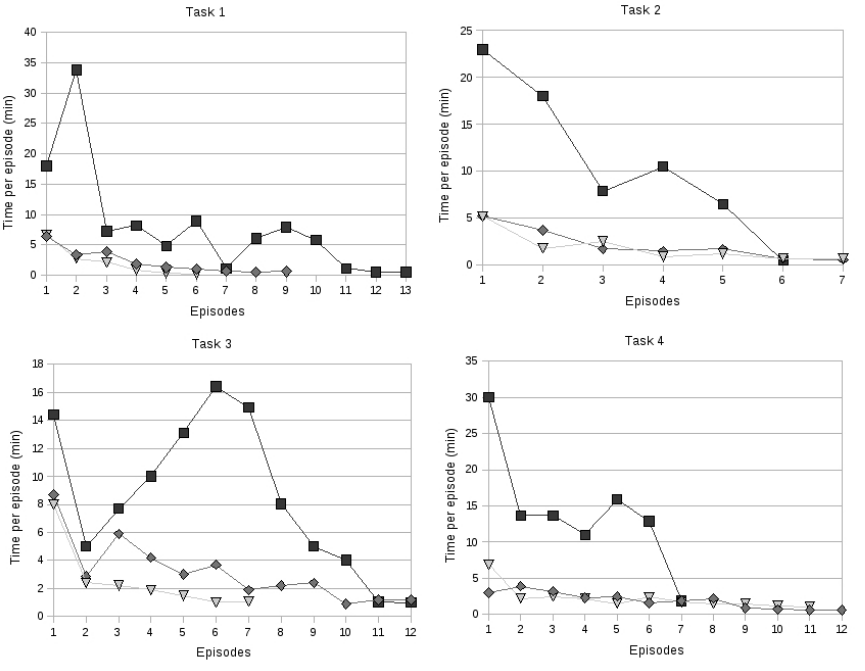


Fig. 2. Average times for learning the four tasks, using different variations of the algorithm: without traces and without feedback (square), with feedback (diamond), with traces and feedback (triangle). The x-axis has the number of episodes and the y-axis has the duration (minutes) of each episode.

Table 3. Number of interventions of the different experiments per task (RL + F = RL with feedback from the user and RL + T + F = RL + F with the initial traces provided by the user) on the left. Times per episode and number of interventions for Task 2 (on the right) both with the initial traces, first with noisy feedback, second with perfect feedback (PF).

	RL + F	RL + T + F
T1	71	23
T2	53	34
T3	143	40
T4	69	72
Avg	84	42

	Times	Num. interventions
T2	13	34
T2 PF	7.6	39

As expected, the learning time is much faster with perfect feedback but also they both have an equivalent number of interventions. This shows that errors in the provided feedback can slow down the learning process but as shown in Table 2, our proposed approach is still between 3 to 6 times faster than normal reinforcement learning and is able to recover from noisy feedback.

As can be seen, by including voice feedback from the user significantly reduces the times of the learning process even with errors in the speech recognition system or in the commands provided. The best results are obtained when the user provides initial traces which also reduces the number of user's interventions.

5 Conclusions and Future Work

In this paper we described a new approach to incorporate feedback from the user into the learning process. The feedback is given by voice commands by a user which are translated into rewards and added to the reward function. Contrary to other approaches the feedback is given at any time during the learning process, can have errors and still converge faster than a traditional reinforcement learning approach. Our approach does not require any special hardware and we believe that it offers a more natural way to train robots new tasks in reasonable training times. We believe that by combining reinforcement learning with on-line feedback from the user is a promising line of research that offers sufficiently fast learning times and a natural instruction mechanism to be used in modern service robots.

There are several areas of improvement and future research work. In particular, we are starting to perform experiments with a manipulator. We would also like to include a command to undo some actions in order to converge faster. Also, we would like to extend our speech recognition system to provide more natural interactions and to consider different intentions, i.e., it is not the same to shout *stop*! to a robot heading towards a staircase than telling gently the robot to *stop* before receiving a new command.

References

1. Abbeel, P., Ng, A.Y.: Apprenticeship Learning via Inverse Reinforcement Learning. In: 21st International Conference on Machine Learning (2004)
2. Conn, K., Peters, R.A.: Reinforcement Learning with a Supervisor for a Mobile Robot in a Real-World Environment. In: IEEE Computational Intelligence in Robotics and Automation (2007)
3. Dorigo, M., Colombetti, M.: Robot Shaping: Developing Autonomous Agents through Learning. Artificial Intelligence Journal 2, 321–370 (1993)
4. Grzes, M., Kudenko, D.: Learning Shaping Rewards in Model-based Reinforcement Learning. In: Workshop on Adaptive Learning Agents ALA-AAMAS (2009)
5. Gullapalli, V.: Reinforcement Learning and its Application to Control. Ph.D. Thesis. University of Masachussetts (1992)
6. Iida, F., Tabata, M., Hara, F.: Generating Personality Character in a Face Robot through Interaction with Human. In: 7th IEEE International Workshop on Robot and Human Communication, pp. 481–486 (1998)
7. Konidaris, G., Barto, A.: Autonomous Shaping: Knowledge Transfer in Reinforcement Learning. In: 23rd International Conference on Machine Learning (2006)
8. Knox, W.B., Stone, P.: Combining Manual Feedback with Subsequent MDP Reward Signals for Reinforcement Learning. In: 9th International Conference Autonomous Agents and Multiagent Systems (2010)

9. Laud, A.: Theory and Application of Reward Shaping in Reinforcement Learning. PhD. Thesis. University of Illinois (2004)
10. Lockerd, T.A., Breazeal, C.: Reinforcement Learning with Human Teachers: Evidence of Feedback and Guidance with Implications for Learning Performance. In: 21st National Conference on Artificial Intelligence (2006)
11. Lockerd, T.A., Hoffman, G., Breazeal, C.: Real-Time Interactive Reinforcement Learning for Robots. In: Workshop on Human Comprehensible Machine Learning (2005)
12. Lockerd, T.A., Hoffman, G., Breazeal, C.: Reinforcement Learning with Human Teachers: Understanding How People Want to Teach Robots. In: 15th IEEE International Symposium on Robot and Human Interactive Communication, pp. 352–357 (2006)
13. Marthi, B.: Automatic Shaping and Decomposition of Reward Functions. In: 24th International Conference on Machine Learning, pp. 601–608 (2007)
14. Mataric, M.: Reward Functions for Accelerated Learning. In: 11th International Conference on Machine Learning, pp. 182–189 (1994)
15. Ng, A.Y., Harada, D., Rusell, S.: Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping. In: 16th International Conference on Machine Learning, pp. 278–287 (1999)
16. Peters, J., Vijayakumar, S., Schaal, S.: Reinforcement Learning for Humanoid Robotics. In: 3rd IEEE-RAS International Conference on Humanoid Robots (2003)
17. Pineda, L.: Corpus DIMEx100 (Level T22). DIME Project. Computer Sciences Department. IIMAS, UNAM, ISBN:970-32-3395-3
18. Randlov, J., Alstrom, P.: Learning to Drive a Bicycle using Reinforcement Learning and Shaping. In: 15th International Conference on Machine Learning, pp. 463–471 (1998)
19. Rybski Paul, E., Kevin, Y., Jeremy, S., Veloso Manuela, M.: Interactive Robot Task Training through Dialog and Demonstration. In: ACM/IEEE International Conference on Human Robot Interaction, pp. 255–262 (2007)
20. Smart, W., Kaelbling, L.: Effective Reinforcement Learning for Mobile Robots. In: IEEE International Conference on Robotics and Automation (2002)
21. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1999)
22. Tenorio-Gonzalez, A.C.: Instruction of Tasks to a Robot using on-line Feedback provided by Voice. M.S. Thesis (2010)
23. Wang, Y., Huber, M., Papudesi, V.N., Cook, D.J.: User-Guided Reinforcement Learning of Robot Assistive Tasks for an Intelligent Environment. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2003)
24. Zhang, Y., Weng, J.: Action Chaining by a Developmental Robot with a Value System. In: 2nd International Conference on Development and Learning (2002)

A Visual Grammar for Face Detection

Augusto Meléndez, Luis Enrique Sucar, and Eduardo F. Morales

Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro No. 1, Tonantzintla, Puebla, Mexico
{amelendez,esucar,emorales}@inaoep.mx

Abstract. Several methods have been developed for face detection with certain success, however these tend to fail under difficult conditions such as partial occlusions and changes in orientation and illumination. We propose a novel technique for face detection based on a visual grammar. We define a symbol relational grammar for faces, representing the visual elements of a face and their spatial relations. This grammar is transformed into a Bayesian network (BN) structure and its parameters are obtained from data, i.e., positive and negative examples of faces. Then the BN is used for face detection via probabilistic inference, using as evidence a set of weak detectors for different face components. We evaluated our method on a set of sample images of faces under difficult conditions, and contrasted it with a simplified model without relations, and the AdaBoost face detector. The results show a significant improvement when using our method.

1 Introduction

Face recognition has been a problem of interest for many years. Several methods for face detection have been developed with certain success, however these tend to fail under difficult conditions such as partial occlusions and changes in orientation and illumination. Under these conditions, it is difficult to distinguish a face in an image. Although the full face is not distinguishable, we can locate some of its elements. Thus, an alternative is to recognize the *basic elements* in a face, and their relationships, and combine this information to build a more robust face detector.

A *visual grammar* can represent the hierarchical decomposition of a face, as well as the spatial and functional relations between its basic elements. Once we represent a face based on a visual grammar, this can be used to *parse* an image and detect the face or faces in it; even if some elements are missing. Although there is some recent work on the development of visual grammars for object recognition [3,11], these are not focused on face recognition. We propose an alternative representation and implementation of a visual grammar, that results in a simpler and more efficient face recognition system.

We propose a novel technique for face detection based on a visual grammar. For this we first define a symbol relational grammar [2] for faces, representing the visual elements of a face and their spatial relations. This grammar is then

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 493–502, 2010.

transformed to a Bayesian network representation. The structure of the Bayesian network is derived from the grammar, and its parameters are obtained from data, i.e., from positive and negative examples of faces. Then the Bayesian network is used for face detection via probabilistic inference, using as evidence a set of weak detectors for different face components.

We evaluated our method on a set of sample images of faces under difficult conditions, and contrasted it with a simplified model without the spatial relations, and also with the AdaBoost face detector. The results show a significant advantage for our visual face grammar in terms of recognition rates, against several variants of the Viola and Jones face detector [7]. The results without including the spatial relations are also inferior, demonstrating the advantage of using a relational grammar.

2 Related Work

2.1 Visual Grammars

Visual grammars entered a hibernation state for some years due to difficulties that remain challenging even today: (i) there is a enormous amount of visual knowledge about the real world scenes that has to be represented in the computer in order to make robust inference, (ii) the computational complexity is huge, and (iii) one cannot reliably compute the symbols from raw images. Recent progress on machine learning, statistical models and more robust visual detectors; as well as the important and steady increase in available computational power; have make possible the rebirth of visual grammars, with several interesting approaches in recent years.

One of the most important contributions is presented by Song-Chun and Munford [11]. In this work they defined a *Grammar of Images* that is represented using an *And-Or Graph*. They propose a common framework for visual knowledge representation and object categorization. Hierarchic and structural composition is the key concept behind these grammars. They discuss three case studies where the principal objective is to represent the structure of the image.

Tian-Fu, Gui-Song and Sng-Chun [8] present a compositional boosting algorithm for detecting and recognizing 17 common image structures in low-middle level vision tasks. This algorithm has two steps: *bottom-up proposal* and *top-down validation*. The bottom-up step includes two types of boosting methods. First, generate hypotheses, often in low resolutions, for some part of the image using AdaBoost, through a sequence of image features. Then propose instances of an image, often in high resolution, by binding existing children nodes of the image. A top-down process validates the bottom-up hypotheses following the Bayesian posterior probabilities.

An attribute graph grammar is defined by Feng Hang and Song-Chu [3]. They used this grammar for image parsing on scenes with man-made objects, such as buildings, kitchens and living rooms. Six rules or productions are defined for the grammar, each production rule not only expands a node into its components, but also includes a number of equations that constrain the attributes of a parent node

and those of its children. The grammar rules are used recursively to produce a large number of objects and patterns in images and thus the whole graph grammar is a type of generative model. The inference algorithm integrates bottom-up detection which activates top-down prediction using the grammar rules.

Previous approaches tend to define very complex models as they intend to represent many classes of objects, and in general this results in complex and inefficient object recognition systems. We propose a more restricted model focused on face recognition, with two important differences to previous models: (i) it is based on a symbol-relation grammar, which provides an adequate formalism for visual grammars by incorporating spatial relations, and (ii) the visual grammar is transformed into a Bayesian network representation that allows to incorporate well known and efficient methods for parameter learning and recognition based on probabilistic inference. Although currently restricted to face recognition, the model can be generalized to represent and recognize other classes of objects.

2.2 Face Detection

There are many approaches for face detection and recognition which we can not cover here in detail due to space limitations (see [10] for a recent survey). Most techniques tend to fail if there are partial occlusions or important variations in illumination and orientation. Most recent approaches for face detection are based in the work of Viola and Jones [7], which combines simple Haar features with a cascade classifier based on the AdaBoost algorithm to detect frontal faces. This results in a very efficient detector with very good results on standard test sets, for example they report 93.9% of correct detections in set CMU+MIT dataset. Note that this dataset has in general frontal images without occlusions. Although superior to previous approaches, these methods are sensitive to face orientation, illumination changes and partial occlusions; under these conditions there is an important degradation in their performance. There are other approaches that consider probabilistic graphical models for face detection [5,1], however these are not based on a visual grammar.

3 A Face Grammar

A common approach to the description of visual languages makes use of formal grammars and rewriting mechanisms able to generate visual sentences. We are using the formalism of *symbol–relation grammars* to define our face grammar.

3.1 Symbol–Relation Grammars

Symbol relation grammars [2] are a syntactic formalism to describe visual languages, where each sentence in a language is represented as a set of visual objects and a set of relationships among those objects. A *Symbol-Relation Grammar* is a 6-tuple

$$G = (V_N, V_T, V_R, S, P, R)$$

where:

- V_N is a finite set of nonterminal symbols.
- V_T is a finite set of terminal symbols.
- V_R is a finite set of (*relation symbols*).
- $S \in V_N$ is the starting symbol.
- P is a finite set of labelled rewriting rules, called *s-item productions* of the form:

$$l : Y^0 \rightarrow < M, R >$$

where:

1. l is an integer uniquely labeling the s-production.
2. $< M, R >$ is a sentence on V_R and $V_R \cup V_T$
   - M is a set of s-tems (v, i), $v \in V_T$ and i is a natural number. In a simple way each s-item is writing v_i.
   - R is a set of r-items of the form $r(X_i, Y_j), X_i, Y_j \in M, r \in V_R$ this indicate a relationship r between X_i and Y_j.
3. $Y \in V_N, Y^0 \notin M$

- R is a finite set of rewriting rules, called *r-items productions*, of the form

$$s(Y^0, X^1) \rightarrow [l]Qos(X^1, Y^0) \rightarrow [l]Q$$

where:

1. $s \in V_R$.
2. l is the label of an s-production $Y^0 \rightarrow < M, R >$
3. $X \in V_N \cup V_T$ and $X^1 \notin M$
4. $Q \neq \phi$ is a finite set of r-items of the form $r(Z, X^1)$ o $r(X^1, Z)$, $Z \in M$

3.2 Relational Face Grammar

Based on the previous formalism we define a relational face grammar. This initial face grammar only considers frontal views and the basic elements of a face; although it could be easily extended to consider other elements (like hair) and other views (like lateral or back). This face grammar includes several spatial relationships which are important because they represent the spatial configuration of the elements of a face, and help to decrease the number of false positives, as *false* basic elements in general do not comply with the desired configuration.

The main elements of this face grammar (FG) are defined as follows:

$$FG = (HEAD, \{eyes, nose, mouth, face\}, \{above, inside\}, HEAD, P)$$

where:

- $HEAD$ is a nonterminal symbol that represents the *complete* face.
- $\{eyes, nose, mouth, face\}$ are the terminal symbol that represent the corresponding elements.
- $\{above, inside\}$ are relational symbols that represent this spatial relations.
- $HEAD$ is the starting symbol

- P are the s-item productions rules, defined as:

  1: $HEAD^0 \rightarrow < HEAD, \phi >$
  2: $HEAD^0 \rightarrow < eyes, \phi >$
  3: $HEAD^0 \rightarrow < nose, \phi >$
  4: $HEAD^0 \rightarrow < mouth, \phi >$
  5: $HEAD^0 \rightarrow < face, \phi >$
  6: $HEAD^0 \rightarrow < \{eyes, mouth\}, above\{eyes, mouth\} >$
  7: $HEAD^0 \rightarrow < \{nose, mouth\}, above\{nose, mouth\} >$
  8: $HEAD^0 \rightarrow < \{eyes, face\}, inside\{eyes, face\} >$
  9: $HEAD^0 \rightarrow < \{nose, face\}, inside\{nose, face\} >$
  10: $HEAD^0 \rightarrow < \{mouth, face\}, inside\{mouth, face\} >$

For this initial face grammar we do not include r-item productions.

In principle, this grammar can be used to parse an image and detect a "HEAD". However, the detectors we used to find the terminal elements are not reliable, and in general there is a lot of uncertainty in detecting the terminal symbols and relations in images. Thus, a representation that can cope with uncertainty is required, such as Bayesian networks [6]. So we transform the relational face grammar to a BN representation.

3.3 Bayesian Networks

Bayesian networks [6] (BNs) are graphical structures that allow for a compact representation of a multi-variable probability distribution. A BN is a directed acyclic graph, $G = (V, E)$, in which the nodes (V) correspond to random variables, and the arcs (E) represent probabilistic dependencies between variables. Associated to each variable their is a conditional probability table (CPT) that contains the probability for each value of the variable given the combinations of values of its parents in the graph (assuming discrete variables). There are several algorithms for learning the structure and parameters of a BN from data, and for estimating the probability of certain variable given a subset of instantiated variables (probabilistic inference) [6,4].

For transforming the face grammar to a BN we make the following considerations:

- Each symbol (terminal or nonterminal) is represented as a binary variable (node).
- Each relation is represented as a binary variable (node).
- Productions are represented as a set of dependency relations (arcs) in the BN. An arc is incorporated between the nodes representing a nonterminal symbol (HEAD) to the terminal symbols. For relations, an arc is added from each of nodes of the symbols in the relation, to the node that represents the relation.

Additionally, we incorporate *virtual* nodes that represent the information we obtain from the detectors for the terminal symbols. These encode the uncertainty inherent to all the detectors as a CPT, and are incorporated to the model by

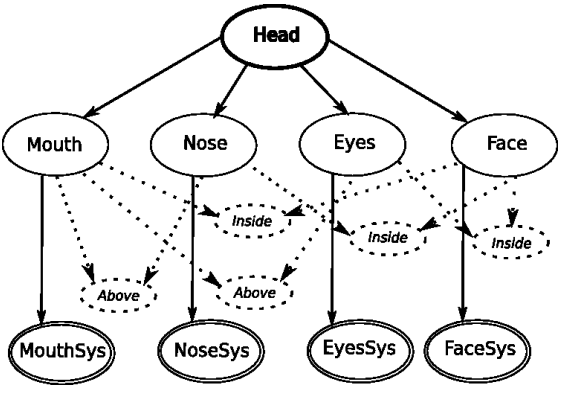


Fig. 1. Bayesian network that represents the face grammar. Simple ovals represent terminal and nonterminal elements, dashed ovals represent the relational elements, and double line ovals represent the detector variables.

adding an arc form the *symbol* node to the *system* node. For instance, we have a node that represents the *nose* symbol, with an arc to a node that represents the results of *nose* detector; which could be a "real nose" or a false detection.

In Figure 1 we depict the Bayesian network that we obtain to represent the face grammar based on the previous considerations. The root node represents the nonterminal symbol $HEAD$. In the next level, there are the nodes that represent the terminal elements of the face; and the nodes that represent the spatial relationship between these elements. In the last level, nodes represent the detectors for each element

3.3.1 Parameter Learning

Once the structure of the BN for the face grammar is defined, we have to obtain the corresponding parameters, i.e. the CPTs for each variable given it parents. We learn these probabilities from data using standard parameter learning for BNs [4].

For this we consider a set of sample images with positive and negative examples of faces. Positive examples are manually labeled indicating the terminal symbols in each image. We then apply the detectors for each terminal element, and count all correct and incorrect detections. Based on these statistics, we obtain the CPTs using a maximum likelihood estimator. Details of this process are given in the experiments.

The parameters for the relational elements could be estimated subjectively, as in general we expect that these relations are always satisfied in a human face. However, there could be special situations in which some relations are not satisfied, for instance if one of the elements is occluded or the face is inclined nearly 90 degrees, so we assigned a probability close to one for each relation being true (these could also be estimated from data but these will require additional effort in the training stage).

3.3.2 Recognition

To recognize a face we have to do a process analogous to parsing for each image using the visual grammar represented as a BN. For this, the terminal elements detected in the image and their spatial relations are instantiated in the BN, and via probabilistic inference we obtain the posterior probabilities of the nonterminal symbol, in this case $HEAD$.

Given that the detectors we use for the terminal symbols produce many false positives, there could be many *candidates* elements to instantiate the BN. In principle, we need to test all possible combinations, which might results in a combinatorial explosion. To avoid this problem, we implement this search process in an *optimal* order, by first testing the more reliable detectors (those that produce less false positives) and then the less reliable ones. Once certain instantiation produces a *high* probability of correct detection (above certain threshold), the search process is terminated; so in this way not all combinations have to be tested. This strategy gave good results in practice, as we will see in the next section.

Additionally we implemented a preprocessing stage to reduce the number of terminal elements detected in an image, as several detections tend to occur that correspond to the same element. If several terminal elements overlap, these were grouped into a single element (depending on the distance between the elements detected).

4 Experimental Results

4.1 Test Images

Although there are several standard face databases, in general they have images of faces in *good* conditions, without occlusions, rotations and other artifacts that make difficult recognition. As our method is focused for these situations, the existing databases are not a good testbed, so we build our own database. For these we downloaded images form Internet that contained faces in a great variety of conditions, most of them difficult for standard detection methods. Some of the images used in the experiments are depicted in Figure 2.

4.2 Terminal Element Detection

For detecting the terminal elements defined in our face grammar, we used detectors based on AdaBoost for faces, nose, mouth and eyes as implemented in the

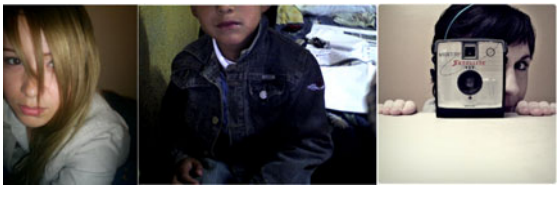


Fig. 2. Examples of face images used in the experiments

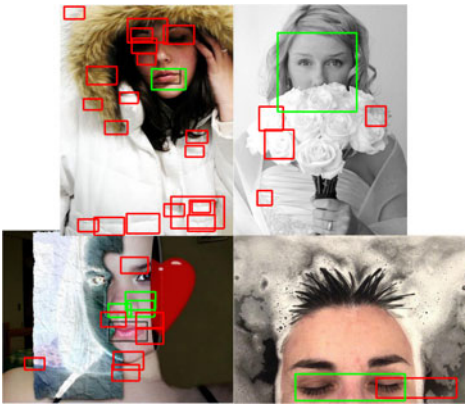


Fig. 3. Examples of the results obtained with different detectors. Green rectangles indicate true positives and red rectangles indicate false positives (best seen in color).

OpenCV library [9]. For the test images considered, these detectors give in general not very good results, tending to produce a large number of false positives, in particular the face, nose and mouth detectors; and also false negatives, specially the eyes detector. Some examples of detections are shown in Figure 3, where we observe a large number of false positives. After applying the preprocessing stage mentioned before to group detections, some elements are eliminated, but there are still several incorrect detections.

4.3 Training the Model

For training we used 200 images, 100 with faces and 100 without faces. For the images with faces with manually labeled the different terminal elements that were visible in the images. Based on these images we obtain the parameters for the BN.

As mentioned before, the CPTs for the relation nodes were estimated subjectively. Different probability values were considered, from 0.6 to 0.9 when the spatial-relation was *true*, and from 0.4 to 0.1 when the spatial-relation was *false*. We selected the values that produced the best results, although the differences are not significant.

4.4 Results

We applied the face grammar BN to a set of images, different from those used from training, 30 positive and 30 negative examples. For each test image, the BN model is applied several times, using the search process described before. The detected terminal elements are instantiated, and then we instantiate the relational nodes. To determinate if the elements detected satisfy the relations that we define, we consider the centroids of the elements detected. Once the

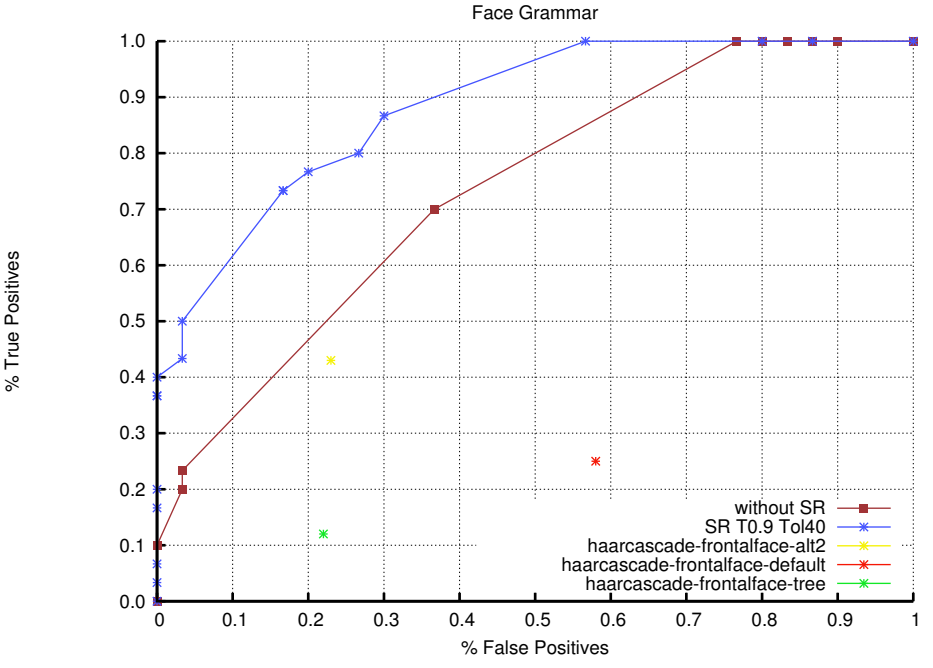


Fig. 4. The graph show the detection rate in terms of true positives vs. false positives varying the decision threshold for the proposed method. We compare the visual grammar with the full model (blue crosses) and the visual grammar with a partial model without relations (red rectangles). We also compare the proposed method against three variants of the Viola and Jones face detector with fixed thresholds (dots).

terminal and relation nodes are assigned, we estimate the probability of *HEAD* using probability propagation.

To evaluate the model we varied the decision thresholds (between 0.5 and 0.85) and compared the true positives vs. the false positives. In Figure 4 we summarize the results. We compared the full model against a partial model that does not include the relations. We observe a significant improvement when spatial relations are incorporated, which supports the choice of a symbol relation grammar. We also compared our method against three variants of the Viola and Jones face detector as implemented in OpenCV [9]. For these other methods we can not control the threshold, so we only show a point result. For the type of images considered, our method clearly outperforms these face detectors.

5 Conclusions

We have developed a novel method for face detection based on a symbol relation grammar for faces. We defined a face grammar that was then transformed to a Bayesian network, whose parameters are obtained from data. We applied this

model for face detection under difficult conditions showing very good results compared against other face detectors.

Although the current grammar is restricted, we consider that this could be extended to provide a more complete description of a head from different viewpoints, and also for representing other classes of objects. In the future we want to explore learning the grammar from examples.

References

1. Candido, J., Marengoni, M.: Combining information in a bayesian network for face detection. Brazilian Journal of Prob. Stat. 23, 179–195 (2009)
2. Ferrucci, F., Pacini, G., Satta, G., Sessa, M., Tortora, G., Tucci, M., Vitiello, G.: Symbol-relation grammars: A formalism for graphical languages. Information and Computation 131, 1–46 (1996)
3. Han, F., Zhu, S.-C.: Bottom-up/top-down image parsing by attribute graph grammar. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1778–1785 (2005)
4. Neapolitan, R.E.: Learning Bayesian Networks. Pearson, Prentice Hall, Chicago, Illiniois (2004)
5. Nefian, A.V.: A hidden Markov model based approach for face detection and recognition. PhD dissertation, Georgia Institute of Technology, Georgia, USA (1999)
6. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco (1988)
7. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57, 137–154 (2004)
8. Wu, T.-F., Xia, G.-S., Zhu, S.-C.: Compositional boosting for computing hierarchical image structures. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
9. Yu, Q., Cheng, H.H., Cheng, W.W., Zhou, X.: Ch opencv for interactive open architecture computer vision. Adv. Eng. Softw. 35(8-9), 527–536 (2004)
10. Zhao, W.Y., Chellappa, R., Philips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Survey, 399–458 (2003)
11. Zhu, S.-C., Mumford, D.: A stochastic grammar of images. Found. Trends. Comput. Graph. Vis. 2(4), 259–362 (2006)

Recognition of Actions That Imply Movement by Means of a Mobile Device with a Single Built-in Accelerometer

César Bernardini

Universidad Nacional de Córdoba
Facultad de Matemática, Astronomía y Física
Haya de la Torre s/n
Ciudad Universitaria
Córdoba, Argentina
cesarbernardini@gmail.com

Abstract. The present document sheds light on a system that recognizes the activities performed by an individual carrying a mobile phone with a built-in accelerometer in the right front pocket of the pants. This accelerometer is used to gather/collect data from certain activities previously established. An analysis of certain learning algorithms is presented as well, containing decision trees, Bayesian method, decision rules, and SVM (Support Vector Machine), all aimed at finding which is better for the recognition of the activities chosen. In addition, certain unsupervised machine learning techniques were used for the analysis of data and actions selected. Results were very favorable, and exhibited a clear distinction of the four (4) types of activities recognized.

Keywords: movement recognition, accelerometer, machine learning, smartphones, mobile computing.

1 Introduction and Motivation

With the passing of time, smartphones get rid of their image of representing an elite device, that only a few people could get access to, and become part of daily stuff/appliances. Statistically speaking, smartphones are said to already have a market penetration which oscillates around 14% to 16%[1]. Smartphones bring about greater computing power to our mobile devices, together with new sensors, which are useful in daily life.

This paper focuses on information that can be acquired by means of cell phones. In addition to storing pictures, SMS, videos, and call recording software, among many other things, cell phones include some other sensors like accelerometers, compass modules, GPS, proximity/orientation sensors, etc. Those sensors make it possible for the user to store information, and, this way, record/track any changes in his/her activities.

[1] http://www.allaboutsymbian.com/news/item/11342_comScore_data_shows_smartphone.php

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 503–511, 2010.

The approach adopted in the present document relies on a mobile system that recognizes user's activities based on data supplied by means of a cell phone accelerometer sensor. Accelerometers are here to stay, and by 2010 it is expected that 33% of cell phones on sale will have one of those sensors[2] incorporated. The recognition of activities can be really useful for different purposes in daily life: controlling diet calories, doing physical activity, changing the profile in the mobile device depending on the activity being performed at a given time, generating playlists to reproduce, implementing game interaction techniques so as to answer questions such as: "How much time I run/walk per day?" "How much time I stay sat when I am working?"

This paper is organized in the following manner. Firstly, some previous work related to the one being presented is described. In Section 3, the system's architecture is explained. Then, in Section 4, the way data were collected/gathered is defined by the detail, in addition to explaining how the data were modified and studied upon by means of unsupervised machine learning techniques. Section 5, on the other hand, presents an analysis of experimental results; and last but not least, a summary containing all the conclusions is provided, where some lines for future work are mentioned.

2 Previous Work

The concept of using accelerometers in order to recognize/detect changes on the immediate context/environment is not new at all. Györbíró, Fábián & Hományi[4] have made use of wireless body sensors in the past, which were attached to some specific parts of human body, so as to detect/catch the activity being performed at a time given: making gestures (body language), walking, running, resting, typing, and cycling.

Soo Yi, Sang Choi & Jacko[7], on the other hand, have employed a single triaxial accelerometer attached to a hand-held device in an attempt to determine what contextual information could be obtained from the acceleration sensor, depending on changes in mobility and light conditions. Bulling, Ward, Gellersen & Tröster [2] have used some accelerometers to detect eye movement and improve a recognition system about what the person is currently doing.

The use of accelerometers is not the only way to detect/perceive motion, Drab & Arter[3] have developed a method for intuitive interaction using optical tracking on mobile devices by resorting to mobile phone video streaming. Anderson and Muller have recognized remaining still, walking and driving with GSM network's information. Buettner, Prasad, Philipose y Wetheral[1] have employed some RFID sensor to detect daily activities as make coffee, watch TV, brush teeth, and some many other. Finally, Peddemors, Eertink & Niemegeers[5] have worked towards a technique to forecast events, which employs retrieved information from visible wireless network entities.

[2] `http://www.cellular-news.com/story/38458.php?source=rss`

All those great papers explain several techniques with external devices, the main objective in this paper is to remove them all, and work only with a simple smartphone carrying an accelerometer sensor.

3 Approach: System Design

By the time being, the system recognizes four actions: *Walking, Remaining still, Running and Remaining inside a vehicle* (for the purposes of this study, and from now on: *Driving*). Our activities recognition system can be classified into the following tasks:

1. Data retrieval.
2. Data preparation.
3. Learning from a classifier.
4. Using the selected classifier to distinguish between different activities.

Data retrieval is done by means of a smartphone and a acceleration sensor. By carrying the mobile phone in the right front pocket of one's pants most of the day, it is possible to record/track information about movement in an ongoing basis. The classifier's learning process has a computational cost too high so as to be executed on the smartphone. Once data retrieval is accomplished, the information gathered is processed, and a classifier is trained in a desktop computer. On the experimental phase, different segmentations of the data were tested, some unsupervised methods were used in order to gain a better understanding of data properties, as well as the relation they bare with different kind of actions; finally, and most importantly, several learning algorithms were drawn in comparison: decision trees, Bayesian method, decision rules & SVM (Support Vector Machine).

After evaluating different approaches whatsoever, the most appropriate classifier was picked and implemented in the smartphone. The said classifier takes as input data from the accelerometer, and then, as an output, it tells us which activity is the person (user) carrying the phone performing.

4 Data and Method

In this section, an explanation is given on how data were collected and made ready; then, a description is provided on the approaches taken towards the systematization of those data with unsupervised and supervised techniques, respectively, which is accomplished by using the Weka machine learning environment [6].

4.1 Data Collection

In order to carry out data retrieval was resorted a smartphone with a built-in accelerometer. The mobile phone used to run the different tests was a Nokia

5800 XpressMusic RM-428[3]. This device incorporates a sensor that makes it possible to calculate the acceleration of the device, device that will then be used to gather information.

Input data that facilitates the learning of the classifier, which recognizes actions, were gathered using the smartphone's accelerometer. This sensor has a coordinates system (Graph), that is used to represent the direction in which acceleration occurs in each axis, as it can be seen on 1 below. Therefore, x-axis and y-axis define a plane, whereas z-axis is perpendicular to the aforementioned plane. The acceleration is positive or negative depending on movement; that is to say, the arrow's direction. Generated accelerations are expressed in $\frac{m}{s^2}$.

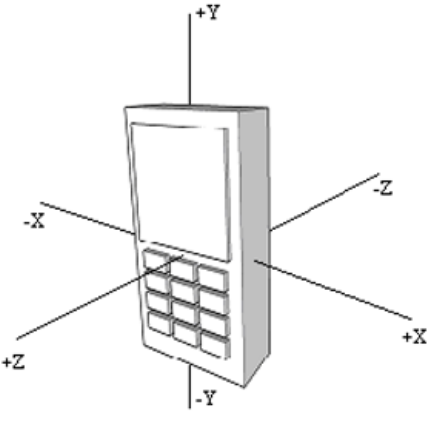


Fig. 1. Smartphone

The accelerometer generates a x, y, z sample by each **0.026**s; that is to say, approximately 38 samples per second. This value was reached using a simple program/software that gathers all the samples that an accelerometer generates within a specific time period. That is, 10 distinct samples were retrieved ranging in duration from 10 seconds to 30 seconds; and after that, the time difference average in which data were collected was calculated.

The *Nokia 5800* device was used on the user's right front pocket of the pants. This specific location was chosen because this is where people usually carry the mobile phone. For the testing process, different types of pants were used: 2 different types of jeans and 2 different types of sweat pants, all of them with front pockets. In accordance, the following samples were gathered from the accelerometer:

- 10 *Walking* samples of 3 minutes of duration each.
- 10 *Running* samples of 3 minutes of duration each.
- 10 *Remaining still* samples of 3 minutes of duration each: in a seated position, lying down, standing up.

[3] `http://developer.symbian.org/wiki/index.php/Nokia_5800_XpressMusic`

– 7 *Driving* samples of 3 minutes of duration each driving in the city;/1 *Driving* sample of 20 minutes of duration in a highway at 120km/h.

From these samples (*Walk, Running, Remaining still & Driving*), one sample of 3 minutes of each type was chosen, and classified accordingly for the test. When all the samples were loaded onto the desktop computer, it was possible to make them ready and consequently apply the machine learning techniques. The samples were segmented into pieces of 1/2, 1, 2, 3, 5, 10 & 15 seconds each.

The objective was to determine, in each of those fragments, the action being carried out. This fragmentation in sampling is done by means of overlapping: a given sample can start at any time, and the number of samples which would occur if using the accelerometer is taken. Once the data were generated and 1/2, 1, 2, 3, 5, 10 & 15 second-samples were achieved, the next step was to analyze them with some of the machine learning tools provided by the Weka environment.

4.2 Exploratory Data Analysis

The aim of these methods is to verify whether the distinction between the different actions that we have chosen to investigate upon are good choices, and emerge naturally from the gathered data. This should serve the purpose of being useful to rule out some of the distinctions; that is or by combining samples or by adding a new distinction among samples.

In these experiments, some clustering techniques were applied to the samples mentioned in previous section. The Weka's k-means algorithm was used with 3, 4 & 5 groups respectively. The evaluation method for the different clustering solutions was the observation of the data distribution among different groups of clusters. On the other hand, the training files for the 2-second-sample were used, because Györbíró, Fábián & Hományi[4] did obtain favorable results with it.

The conclusions from the experiments were:

– *Running* samples should clearly be apart in a different group.
– 19% of *Remaining still* samples were incorrectly classified as *Walking* samples.
– When the amount of groups was reduced, *Remaining still* and *Driving* samples were fused together. It makes sense in a way, since *Remaining still* and *Driving* are actions with little movement.
– When one more group was added (making it 5 groups in total), the Remaining still sample was the one divided in more parts: one part was classified as Driving samples, others were classified as Walking samples, and the rest stood separately.

4.3 Learning a Classifier

A comparison was drawn between different classifiers with several data fragments varying in sizes so as to find out which was the best classifier when it comes to distinguishing between actions.

Several tests were carried out, with decision trees, Bayesian method, decision rules and SVM all being used. All the tests were performed using the basic parameters provided within the Weka Toolkit 3.6.

In order to determine whether the evaluation method produced favorable results, the percentage of samples correctly classified/recognized in a sample not seen before achieving more than 95% was established as "evaluation criteria", and some observations about kappa statistics were made over the 10-fold cross-validation realm.

5 Analysis of Results

Decision trees were the first ones being subjected to testing, process by means of which files were created with all the samples obtained, where, in addition, Weka was allowed to randomly select values so as to make a 10-fold cross-validation. As it can be clearly seen in Graph 1, the results obtained with decision trees were quite correct, if you will, always leading to such percentages oscillating between 99% and 100% in kappa statistics for C-V. On the other hand, it is worth taking into consideration that the values were pretty much constant/coherent, and it was of no importance whether the samples being considered for the study were of one (1) or fifteen (15) seconds respectively: results remained similar in characteristics. This is of true value and importance for the testing process, because whenever it should be necessary to analyze 15-second samples, it could be possible to simply use the first second of the sample, with the results being similar one to each other.

When the samples were subjected to NaiveBayes algorithm, values were not that favorable as the ones obtained before, and oscillated between 88% and 92%, respectively. It should be considered that results of less than one (1) second, were a little bit less effective than the rest, making it evident that in case of using this method, one (1) second samples must be used. The analysis of results within the confusion matrix made it possible to realize that most of the confusion lies between *Walking* and *Remaining still* samples, and also between *Remaining still* and *Driving* samples. Therefore, 19% of *Walking* samples were incorrectly recognized. The best classification corresponds to the *Running* samples, which in 1-second samples exhibited an error margin of 1%.

Decision rules, as in the case of Bayes, did not provide as favorable results as the ones obtained with decision trees, and these varied as to the efficacy degree in 90% within kappa statistical values. Classification errors were quite different in nature, where the best classification results correspond to the *Driving* samples, but with the presence of false positives rounding 5%. As it comes to use, it should be taken into consideration that being able to determine if a sample belongs to an action is, if you will, economical, but it requires that large amounts of memory be used, which must be busy at all times.

Last but not least, when using the Support Vector Machine, values rounding 92% and over 96% were obtained. Classifications belonging to *Driving*, *Running*

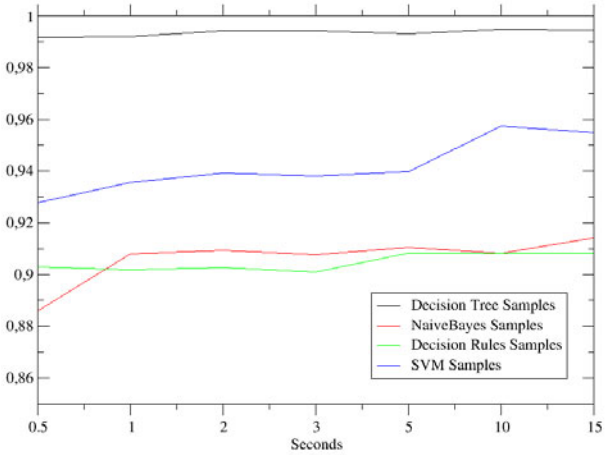


Fig. 2. Results

and *Walking* samples were pretty much correct, with an error margin of less than 1%. This algorithm confusion consisted in *Remaining still* samples being classified as *Walking* samples, with an error margin of practically 20%.

The Graph 2 below shows the values within kappa statistics in regards to 10-fold cross-validation for each of the algorithms used, and for each of the samples employed.

Considerations being drawn into play, it results evident how kappa statistics in the decision tree realm are the highest in value. Also worth of notice is what was mentioned before as to the efficacy escalating 90% within Bayesian method and decision rules, respectively.

Once these analyses were done, the decision tree was deployed so as to test the samples that, at first, had been driven apart. Results exhibited a 99% efficacy level for all and each one of the samples.

6 Conclusions and Future Work

This work taken as a base makes it possible to draw some conclusions on the basis of the results that were obtained:

– Decision trees provided the best results in the set, whose efficacy did not present as many variations regarding the amount of data collected. Therefore, it could be said that considering one (1) second, data will show results similar in characteristics as 15-second data, which entitles less data processing (calculation) being performed and less memory being used. It was the algorithm the one method that diverged most from the rest and exhibited the best results.

- Sample collection was performed with more volunteers so as to determine if the understanding gained from one person can be obtained from the same activities carried out by a different person, or else this person has to undergo the same training process done here.
- Running samples were the most easily recognized samples by all the algorithms used, which makes it a real possibility that in case it was necessary to implement some kind of a system able to perform actions when an individual is running, very good results would be obtained.
- More consistent analyses of battery usage and optimization of data processing for the work in a mobile device.
- The bigger problems arose because of the wrong classification of *Remaining still* samples in three (3) of the four (4) algorithms used.

All this constitutes a precedent so as to carry on the work and engage in certain interesting experiences. For example:

- Performing the same procedure but adding some other kind of actions: *Going up the stairs, Going down the stairs, Remaining still (Standing up, Lying down), Jumping, Using an elevator, Spinning.*
- Adding layers of intelligence: layers of intelligence should analyze the results of the samples being considered and, on the basis of certain heuristics, it should be decided whether the change from one action to the other is viable, or if this is simply a classification error.
- Using some other sensor so as to determine the location of the mobile phone and trying to correct data to render a correct classification of the sample in question. Integration of GPS, or some indoor/outdoor localization system, and intelligence layers may lead to some very interesting uses of the system.

Acknowledgments

The author would like to thanks Mr. Felipe Manzano for his constant help, which was very usefull while doing this work. He would also like to thank, his Artificial Intelligence teacher, Dr. Laura Alonso Alemany, for her motivating attitude towards this job. And last but not least, he would like to thank Mrs. MarÃa Laura Granollers for her translation job.

References

1. Buettner, M., Prasad, R., Philipose, M., Wetherall, D.: Recognizing daily activities with rfid-based sensors. In: Proceedings of the 11th International Conference on Ubiquitous Computing, Ubicomp 2009, pp. 51–60. ACM, New York (2009)
2. Bulling, A., Ward, J.A., Gellersen, H., Tröster, G.: Eye movement analysis for activity recognition. In: Proceedings of the 11th International Conference on Ubiquitous Computing, Ubicomp 2009, pp. 41–50. ACM, New York (2009)
3. Drab, S.A., Artner, N.M.: Motion detection as interaction technique for games & applications on mobile devices. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 48–51. Springer, Heidelberg (2005)

4. Györbíró, N., Fábián, A., Hományi, G.: An activity recognition system for mobile phones. Mob. Netw. Appl. 14(1), 82–91 (2009)
5. Peddemors, E.y.N.: Predicting mobility events on personal devices
6. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and te chniques. Morgan Kaufmann, San Francisco (2005)
7. Yi, J.S., Choi, Y.S., Jacko, J.A., Sears, A.: Context awareness via a single device-attached accelerometer during mobile computing. In: Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services, MobileHCI 2005, pp. 303–306. ACM, New York (2005)

Development of a Tour–Guide Robot Using Dialogue Models and a Cognitive Architecture

Héctor Avilés, Montserrat Alvarado-González, Esther Venegas, Caleb Rascón, Ivan V. Meza, and Luis Pineda

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS)
Universidad Nacional Autónoma de México (UNAM)
{haviles,mon}@turing.iimas.unam.mx, esthervenegas@hotmail.com,
{caleb,ivanvladimir}@turing.iimas.unam.mx, luis@leibniz.iimas.unam.mx

Abstract. In this paper, we present the development of a tour–guide robot that conducts a poster session through spoken Spanish. The robot is able to navigate around its environment, visually identify informational posters, and explain sections of the posters that users request via pointing gestures. We specify the task by means of dialogue models. A dialogue model defines conversational situations, expectations and robot actions. Dialogue models are integrated into a novel cognitive architecture that allow us to coordinate both human–robot interaction and robot capabilities in a flexible and simple manner. Our robot also incorporates a confidence score on visual outcomes, the history of the conversation and error prevention strategies. Our initial evaluation of the dialogue structure shows the reliability of the overall approach, and the suitability of our dialogue model and architecture to represent complex human–robot interactions, with promising results.

Keywords: Dialogue systems, service robots, human–robot interaction, pointing gestures.

1 Introduction

The development of autonomous service robots has received considerable attention in the past decade. These robots should be capable to engage in meaningful, coherent, task–oriented multimodal dialogues with humans to carry out daily–life activities. In addition, these robots must also include a variety of perceptual and behavioural abilities to effectively achieve their goals. When designing such robots, it is desirable to consider conceptual frameworks for the description of the tasks in which domain knowledge, interaction structures, and robot capabilities are integrated in a simple, comprehensible, and coordinated form.

For this purpose, we present the development of a tour–guide robot using dialogue models (*DMs*) [1]. In our context, a DM is a highly abstract interaction protocol, stated declaratively at an intentional level, that allow us to easily specify robot tasks, independently of the input and output capabilities of the robot. DMs are integrated into a novel cognitive architecture that fully coordinates HRI and robot capabilities [2]. Along with DMs, this architecture offers a

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 512–521, 2010.

flexible development framework to design complex tasks for service robots [3]. Our robot is able to conduct a poster session in which speech and pointing gestures can be used by human users to further along the interaction. As part of the tour, the robot visits different posters by request of the user through the scenario. The dynamics of the multimodal interaction is enriched by the use of confidence values and the history of the conversation. Moreover, the flow of the conversation is aided by error prevention and recovery strategies. To evaluate this proposal, we performed an initial evaluation of the overall functionality of the system, the interaction modalities, and the user satisfaction, using a real robot. Our results show that the robot accomplished the task the majority of the time and that the users were satisfied with the experience.

The rest of the paper is organized as follows. Section 2 introduces previous proposals on service robots closely related to our work. A brief description of the cognitive architecture and dialogue models is presented in Section 3. Section 4 is devoted to the specification of the tour–guide task. Section 5 describes the algorithms that support language, vision and motor behaviors of our robot. Section 6 presents the evaluation procedure and results. Finally, conclusions and future work are discussed in Section 7.

2 Related Work

One of the first service robots with multimodal communication capabilities is ROBITA [4]. This robot answers questions in spoken Japanese about the location of a person in an office environment using speech and pointing gestures. ALBERT [5] grasps objects following speech and hand posture commands of the type "Hello Albert, take this". Jido [6] tracks head and hands of a user, and answer requests such as "Come here". Markovito [7] recognizes spoken commands in Spanish, delivers objects and spoken messages, and also identifies nine different types of gestures. Notwithstanding the advanced capabilities provided by these robots, none of these examples provides an integration framework focused on the interaction between the robot and the user.

Other approaches have applied dialogue models to describe the task. An example is BIRON [8], a robot that uses pointing gestures and visual information of the environment to resolve spoken references to objects. In [9] a robot controlled by a dialogue system based on simple finite state machines is presented. The dialogue system of the Karlsruhe humanoid robot [10] coordinates object manipulation, localization, recognition and tracking of a user, recognition of gaze and pointing gestures, and interactive learning of dialogues. However, the majority of these examples were build focused on a specific task which complicates the generalization of their frameworks. The overall approach presented in this document is a contribution to solve these problems.

3 A Cognitive Architecture Oriented towards HCI

The cognitive architecture that we use is a three–layer structure focused primarily on multimodal human–computer interaction (*HCI*) –See [2] for a graphical

description of this architecture. Perception and action processes are arranged into two main branches that we labeled *recognition and interpretation* and *specification and rendering*.

The recognition and interpretation branch provides an abstract interpretation of the world accordingly to the *expectations* defined by the representation/inference layer –described below. This branch process inputs of different perceptual modalities, and there are as many instances of this branch as required. Each instance specializes in a single perceptual capability of the robot. The specification and rendering branch corresponds to behavioral responses. This branch provides the means in which the robot can influence and act over the world. Similar to the recognition and interpretation branch, there could be as many instances as needed.

The top layer of the architecture coordinates the processes described above, and also keeps track of the interactions between the robot and its environment. This layer is specified by a dialogue model, that is a set of expected multimodal *situations* which defines the interaction protocol. A DM can be described as a directed graph, in which situations are represented as nodes, and edges are labeled with expectation–action pairs. Whenever an expectation is met, the corresponding action is triggered. A simple example is depicted in Fig. 1. This example shows a situation S_i, with an edge $\alpha : \beta$ that is directed to the next situation S_j. Situations can have one or more input and output expectation–action pairs, and are typed according to the modality of the expected input; the main types are *listening* –or linguistic– and *seeing* –or visual. There is also a special class of situations called *recursive*, in which a full DM is embedded, and allows to modularize the specification. All DMs have one initial and one or more final situations. DMs are based on *recursive transition networks*, augmented with functions standing for expectations, actions and next situations [1].

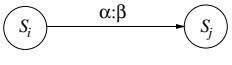


Fig. 1. Situations in a dialogue model

This cognitive architecture not only provides a "natural" representation of the flow of different HCI tasks [11], but it also simplifies the integration of perceptual and behavioural algorithms –independently of their implemented modality– through the definition of the expectations. This abstraction could be particularly important whenever the system process inputs perceived by different modalities at the same time.

4 Specification of the Task

The dialogue model developed the tour-guide robot application is described as follows. Our main dialogue model M consists of an initial greeting situation S and four recursive situations which direct the execution of the task: a)

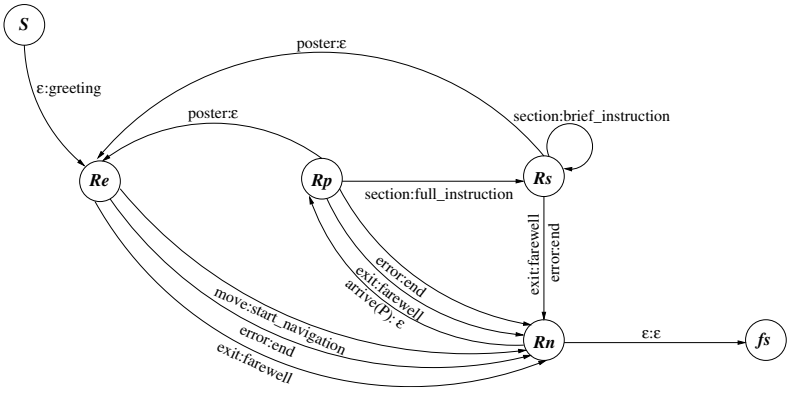


Fig. 2. Main dialogue model

the choose–poster sub-dialogue *Re*, b) the see–poster sub-dialogue *Rp*, c) the pointing–section sub-dialogue *Rs*, d) the move–to–poster sub-dialogue *Rn*, and e) the final situation *fs*. The main dialogue model is shown in Fig. 2.

Situation *S* is triggered by the execution of the program. The robot verbally presents itself and explains the goal of the task. Sub–dialogue *Re* leads the robot to ask the user to choose a poster from a list of available posters –this list is dynamically built depending on the history of the conversation. The robot listens for the poster selected by the user. Then, it navigates around its environment from its initial point to the fixed coordinates of the poster. Once there, it announces its arrival and indicates that it will visually recognize the poster –situation *Rn*. The visual module evaluates the quality of the recognition of the poster and gives the result to the *Rp* sub–dialogue. If the evaluation is high, the dialogue firmly states the poster, if it is medium, the dialogue is rendered as if the robot has some uncertainty on the matter. The robot then offers the user to explain a section of the current poster or to visit another one. If the quality is low or the robot did not see the poster, then the dialogue expresses doubts about the quality of the recognition and only offers the user to visit another poster. The previous design principle, as part of an adequate specification, is an effective error prevention strategy. If the user requires an explanation of a section of the poster, the robot asks him/her to point to the desired text or image, for some seconds over the section, after a beep sound. Then, the visual module evaluates the quality of the identification of the section and gives the result to the *Rs* sub–dialogue. The degree of certainty of the identification is mentioned during the conversation. Then the robot explains the section captured and offers the user to explain another section of the current poster or to visit another one. Our task specification also include recovery strategies for speech interpretation misunderstandings. In these cases, the system provides the appropriate feedback the user in order to proceed with the interaction.

Table 1. Example of the interaction between the robot and a user; robot and user actions are indicated in italics

Turn	Actor	English Translation
1	ROBOT	Hi, I am Golem. I can describe information about posters, and also its sections. We have the posters of "Golem", "Mesoamerica" and "Encryption". Which one do you want me to explain?
2	USER	I want to see the Golem poster.
3	ROBOT	Ok, let's go to the Golem poster. *The robot moves from its initial position to the poster.*
4	ROBOT	I've arrived to the Golem poster. Please, hold on a second while I confirm if it's the correct poster. *The visual module indicates that the quality of the recognition of the poster is medium.*
5	ROBOT	I'm not sure, but I think that it's the Golem poster. Do you want me to explain a section or do you want to see another poster?
6	USER	No user response.
7	ROBOT	Sorry, I did not hear you.
8	USER	Please, explain a section.
9	ROBOT	Ok, please point to the desired text, image or section, by making a soft movement with your hand for some seconds over it, after the beep sound. *The user do nothing, and the visual module indicates that the quality of the identification is low.*
10	ROBOT	I could not see you pointing at any region, maybe because you did not point or because I did not see the region you pointed at. Do you want me to explain a section or do you want to see another poster?
11	USER	Explain a section.
12	ROBOT	Ok, don't forget to point after the beep.
13	USER	*The user points to a picture of the robot.*
14	ROBOT	*The quality of the of the identification is high.* It's me! Do you want me to explain a section or do you want to see another poster?
15	USER	I want to finish.
16	ROBOT	Ok, thank you for your visit, goodbye!

Any time the user wants the robot to finish the task or whenever any of the DMs indicate a major error, the robot returns to its initial position and the dialogue ends –situation *fs*. Table 1 shows an example of an interaction between the robot and a user.

5 Robot Capabilities

In this section we describe the visual, speech and navigation capabilities of our robot. The three modules that perform these procedures are herewith explained.

5.1 Visual Module

This module is in charge of two main tasks: a) gesture analysis and b) poster recognition. Pointing gesture analysis involves two main steps: i) 2D arm segmentation, and ii) identification of the section pointed by the user. For segmentation, the arm is spotted into a binary image by fusioning motion data, and the absolute difference of the first view of the poster in the actual robot position and the current image in the stream. Data fusion is performed by following a simple logic AND rule.

Simple region growing techniques are used to segment the arm. Least–squares method is used to fit a line to the arm and its tip is selected by comparing the distance from both extreme points of the line to the vertical edges of the poster. Arm segmentation lasts a maximum of 10 seconds. If a section has not been reliably identified at the end of that period, a sub–dialogue with the user starts.

Poster recognition is achieved by the SIFT algorithm. The poster with maximum number of matches and above an empirical threshold corresponds to the classification result. If an appropriate quality in recognition is achieved, the vision agent receives the coordinates of the rectangle that delimits each relevant section of the poster. To adjust these coordinates to the actual view, we calculate a projective transformation matrix using SIFT matches and RANSAC algorithm. Once all visible sections have been calculated, a rectangular window is defined to enclose these sections. An extended explanation of the visual module can be found in [12].

5.2 Speech Understanding Module

The Speech Understanding module performs three main functions: i) speech recognition, ii) speech interpretation, and iii) speech synthesis, in Spanish. Speech recognition is based on the DIMEx100 project [13]. This project is aimed to develop general acoustic–phonetic models and pronouncing dictionaries in Spanish for Mexican people of different gender and age. Synthesis of spoken dialogue of the robot is executed with MBROLA synthesizer [14].

Every time a sentence is recognized, speech interpretation takes place. This is done by comparing the sentence with a set of equivalent regular expressions defined for each expectation. If no match is found, the user is warned and requested for another attempt.

5.3 Navigation Module

Our robot navigates in a 2D world. The dialogue manager informs to the agent the target position (x, y) of the poster as well as θ, the final angular orientation.

First, the robot rotates to reach the orientation of the target, and moves along a straight line up to this coordinate. We assume there are no obstacles along this path. Once the robot has arrived, it rotates again up to reach the final θ orientation. Robot motion is measured using odometry only.

6 Experiments and Results

In this section, we present a preliminary round of tests performed to evaluate the overall functionality of our proposal and its results. First we describe our experimental setup.

6.1 Experimental Setup

Our robot is an ActivMedia's Peoplebot robot with a Canon VCC5 camera. A 2Gb 1.86GHz Dual–Core laptop computer is attached to the robot through a wired Ethernet connection to distribute processing load. Image frame rate is about 30 FPS. Robot motors and sensors are controlled using Player/Stage libraries. SIFT templates of the posters were taken with the camera at a distance of 1m, and a height of 90cm. The distance threshold to evaluate square Euclidean distances between two SIFT features is fixed to 0.2. The minimum number of SIFT matches to consider a positive classification result is 50. The size of each poster is $1.5 \times .70$m. A cheap headset is plugged to the internal sound card of the laptop and the synthesised speech is played through the robot speakers. We used our lab environment with artificial light. Three informational posters were positioned in a U-shaped distribution, located at a distance 2.20m between them.

6.2 Evaluation and Results

To evaluate our approach, we defined the tour–guide task as follows. The robot must: i) initiate the interaction with a single user, ii) explain at least one poster and one section selected by the user, and, finally, iii) finish the interaction going back to a predefined initial position. We asked 10 participants to interact with the robot –one at a time– to accomplish this task. Participants are either students or members of the academic staff of our department, each of whom have different levels of expertise in robotics. At the beginning of the exercise, users were advised to follow the instructions provided by the robot, and no special recommendations were given afterwards. Figure 3 shows an example of an interaction. In this case, the user indicates one section of the poster to get further information. At the end of the exercise, each user was requested to fill out a questionnaire that surveys two traditional metrics: the *effectiveness* of the system and the *satisfaction of the user*. Effectiveness measures the accomplishment of the task and the performance of the system modules. Satisfaction ratings assess the perception of the user about the system.

On the one hand, the evaluation of effectiveness shows that 9 out of 10 participants fulfilled the goal. Pointing gestures were successfully recognised 88.9% of

Fig. 3. Our tour–guide robot in action. In this example, the user is pointing to a section of the poster. In the background of the scene, our evaluator takes notes of the robot's performance.

Table 2. User satisfaction results

Topic	4	3	2	1
TTS Easy to understand	8	2		
ASR System understood	5	3	2	
Task easy	5	3	2	
Task easy poster names	9	1		
What to say	9	1		
Expected behaviour	4	6		
Future use	10			
Naturalness pointing	2	6	2	
Naturalness conversation	2	8		

the time and the posters were always identified correctly; 60% with high confidence and 40% with medium confidence. The robot arrived to the position of the desired poster for all interactions. From the speech recognition side, the request to explain a poster was correctly understood 90% of the time, while the request for a section, 80%. On the other hand, the evaluation of satisfaction included questions from the PARADISE framework regarding *text-to-speech* and *automatic speech recognition* quality, ease of use, user experience, expected behavior and future use [15]. We also considered two additional questions regarding the naturalness of both, the spoken conversation and the pointing gestures. Table 2 shows a summary of the results[1]. In this experiments, all the participants continued interacting with the system beyond the specified goal. We observed that all participants are willing to interact with the system in the future. However, a main issue we are interested in is improving the naturalness the execution of gestures and the spoken conversation.

[1] Users were forced to measure either a positive or a negative response, mapped to a rate from 1 to 4, where 4 is the highest score.

7 Conclusions and Future Work

We presented our work on the development of a tour–guide robot using dialogue models. Dialogue models coordinate the human–robot interaction using speech understanding, visual analysis and navigation. Our robot is able to navigate in its environment, and to describe informational posters and sections selected by the user using pointing gestures. The robot also evaluates confidence in its visual results, considers the history of the conversation and modify its dialogues accordingly. Our results showed the effectiveness of the overall approach: all users are willing to interact with the robot in the future and the user satisfaction ratings are, in general, positive.

As future work we plan to refine our dialogue model as well as the effectiveness of the speech and visual modules. In addition, we will consider speech and gestures entries in a multimodal manner. We are going to implement 3D pointing gestures using a binocular stereo system. Finally, we will conduct an in–depth study about the relations of speech and gesture to incorporate common grounding, reference resolution and history management into the actual system using these two input modalities.

Acknowledgments. This paper has been funded by research grants from CONACyT, Grant 81965, and from PAPIIT–UNAM, Grant 104408.

References

1. Pineda, L.A.: Specification and Interpretation of Multimodal Dialogue Models for Human-Robot Interaction, In: Sidorov, G. (ed.) Artificial Intelligence for Humans: Service Robots and Social Modeling, pp. 33–50. Sociedad Mexicana de Inteligencia Artificial (2008)
2. Pineda, L.A., Meza, I., Salinas, L.: Dialogue Model Specification and Interpretation for Intelligent Multimodal HCI. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 20–29. Springer, Heidelberg (2010)
3. Rascón, C., Avilés, H., Pineda, L.A.: Robotic Orientation towards Speaker in Human-Robot Interaction. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 10–19. Springer, Heidelberg (2010)
4. Tojo, T., Matsusaka, Y., Ishii, T.: A Conversational Robot Utilizing Facial and Body Expressions. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 858–863 (2000)
5. Rogalla, O., Ehrenmann, O., Zoellner, M., Becher, R., Dillmann, R.: Using gesture and speech control for commanding a robot assistant. In: 11th IEEE Workshop on Robot and Human Interactive Communication, pp. 454–459 (2002)
6. Burger, B., Lerasle, F., Ferrane, I., Clodic, A.: Mutual Assistance between Speech and Vision for Human–Robot Interaction. In: IEEE/RSJ International Conference on Intelligent Robotics and Systems, pp. 4011–4016 (2008)
7. Aviles, H., Sucar, E., Vargas, B., Sanchez, J., Corona, E.: Markovito: A Flexible and General Service Robot. In: Liu, D., Wang, L., Tan, K.C. (eds.) Studies in Computational Intelligence, vol. 177, pp. 401–423. Springer, Heidelberg (2009)

8. Toptsis, I., Haasch, A., Hüwel, S., Fritsch, J., Fink, G.A.: Modality Integration and Dialog Management for a Robotic Assistant. In: European Conference on Speech Communication and Technology, pp. 837–840 (2005)
9. Lee, C., Cha, Y.S., Kuc, T.Y.: Implementation of Dialog System for Intelligent Service Robots. In: International Conference on Control, Automation and Systems, pp. 2038–2042 (2008)
10. Stiefelhagen, R., Ekenel, H.K., Fugen, C., Gieselmann, P., Holzapfel, H., Kraft, F., Nickel, K., Voit, M., Waibel, A.: Enabling Multimodal Human–Robot Interaction for the Karlsruhe Humanoid Robot. Trans. on Robotics: Special Issue on Human–Robot Interaction 23(5), 840–851 (2007)
11. Meza, I., Salinas, S., Venegas, S., Castellanos, H., Chavarria, A., Pineda, L.A.: Specification and Evaluation of a Spanish Conversational System Using Dialogue Models. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 346–355. Springer, Heidelberg (2010)
12. Aviles, H., Meza, I., Aguilar, W., Pineda L.: Integrating Pointing Gestures 2nd International Conference on Agents and Artificial Intelligence, pp. 585–588 (2010)
13. Pineda, L., Castellanos, H., Cuétara, J., Galescu, L., Juárez, J., Llisterri, J., Pérez, P., Villaseñor, L.: The corpus dimex100: Transcription and evaluation. Language Resources and Evaluation (2009)
14. Dutoit, T., Pagel, V., Pierret, N., Bataille, F., Van Der Vrecken, O.: The MBROLA Project: Towards a Set of High-Quality Speech Synthesizers Free of Use for Non–Commercial Purposes. In: 4th International Conference on Spoken Language Processing, vol. 3, pp. 1393–1396 (1996)
15. Walker, M., Litman, D., Kamm, C., Kamm, A.A., Abella, A.: Paradise: A framework for evaluating spoken dialogue agents. In: 35th Annual Meeting of the Association for Computational Linguistics, pp. 271–280 (1997)

Detection of Multiple People by a Mobile Robot in Dynamic Indoor Environments

José Alberto Méndez-Polanco, Angélica Muñoz-Meléndez, and Eduardo F. Morales-Manzanares

National Institute of Astrophysics, Optics and Electronics, Computer Science Department, Luis Enrique Erro 1, 72840 Tonantzintla, México
{polanco,munoz,emorales}@inaoep.mx

Abstract. Detection of multiple people is a key element for social robot design and it is a requirement for effective human-robot interaction. However, it is not an easy task, especially in complex real world scenarios that commonly involve unpredictable motion of people. This paper focuses on detecting multiple people with a mobile robot by fusing information from different sensors over time. The proposed approach applies a segmentation method that uses the distance to the objects to separate possible people from the background and a novel adaptive contour people model to obtain a probability of detecting people. A probabilistic skin model is also applied to the images and both evidences are merged and used over time with a Bayesian scheme to detect people. We present experimental results that demonstrate how the proposed method is able to detect people who is standing, sitting and leaning sideways using a mobile robot in cluttered real world scenarios.

1 Introduction

In recent years significant progress has been achieved in the field of service robots, whose goal is to perform tasks in populated environments, such as hospitals, offices, department stores and museums [3]. In this context, people detection by a mobile robot is important because it can help to improve the human robot interaction, perform safety path planning and navigation to avoid collisions with people, search lost people, recognize gestures and activities, follow people, and so on.

A previous work[2] presents a people detection method which took into account the distance between the robot and people to try to get the best fit model. Therefore, the previous work detected people without assuming that the person is facing the robot. However, the main drawback of that method was the number of false positives detected. In this work, we extend our previous work to: (i) detect multiple people with a mobile robot, (ii) fuse information from different sensors over time, and (iii) apply a Bayesian scheme without a prior visual information about the environment. The sensor fusion significantly reduced the number of false positives in the detection.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 522–531, 2010.

The paper is organized as follows: In Section 2 we present related work. Section 3 describes our segmentation method based on the distance to detected objects. Section 4 introduces the adaptive contour model to get a first estimation of the people position relative to the robot. Section 5 presents the skin detection method. Section 6 shows the people detection and tracking algorithms. Section 7 shows the experiments and results. Finally, Section 8 presents the conclusions and future research directions.

2 Related Work

Depending on the specific application that integrates people detection and identification there are two principal approaches that can be applied; whole human body detection [4,5] and part-based body detection [6] over images acquired, for instance, with single or stereo charge coupled device (CCD) camera and thermal cameras.

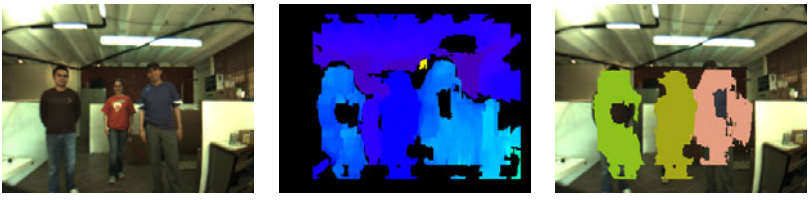
The advantages of whole human body detection approaches are the compact representation of the human body models such as human silhouettes [4,6]. The main drawbacks of these approaches are the sensitivity to object occlusion and cluttered environments as well as the need of robust object segmentation methods. Concerning part-based body detection approaches, the main advantage is their reliability to deal with cluttered environments and object occlusion because of their approach to detect human body parts. In contrast to whole body detection, part-based body detection approaches do not rely on the segmentation of the whole body silhouettes from the background. Part-based body detection approaches, in general, aim to detect certain human body parts such as face, arms, hands, legs or torso. Different cues are used to detect body parts, such as laser range- finder readings to detect legs [7] or skin color to detect hands, arms and faces [8]. Although people detection using static cameras has been a major interest in computer vision [9], some of these works cannot be directly applied to a mobile robot which has to deal with moving people in cluttered environments.

3 Distance Based Segmentation

We use our the proposed method in [2] to segment the information provided by a stereo camera. The distance to the objects can be calculated with an adequate calibration of the stereo camera and using a disparity image. Once the distance to the objects has been calculated, we scan one of the stereo camera images to segment objects based on the obtained depth information. The idea is to form clusters of pixels with similar distances to the robot (Z coordinate) and, at the same time, near each other in the image (X and Y coordinates). The clusters are defined as follows:

$$C_k = \{\mu_X^k, \mu_Y^k, \mu_Z^k, \sigma_X^k, \sigma_Y^k, \sigma_Z^k, \rho_k\}, k = 1...N \quad (1)$$

where μ_X^k, μ_Y^k and μ_Z^k are the means of the X,Y,Z coordinates of the pixels within the cluster k, σ_X^k, σ_Y^k and σ_Z^k are the variances of the X,Y,Z coordinates



(a) Original image (b) Disparity image (c) Segmented image

Fig. 1. Segmentation method. These examples illustrate how our segment method is able to segment one or multiple people. (a) is the original image, (b) is the disparity image and (c) is the result of our segmentation process.

of the pixels within the cluster k, ρ_k is a vector containing the coordinates of the pixels in the cluster k, and N defines the maximun number of clusters.

After segmentation, clusters with irregular proportions ($\mu_X > \mu_Y$) are eliminated to take into account only objects with human proportions.

Figure 1 shows examples of the object segmentation based on distance using the method described in this section. In this example the background is discarded because it has no enough depth information so, any cluster is created for this region. This example illustrates the reliability of our method to segment simultaneously one or more people.

4 Adaptive Semi-elliptical Contour Model

The segmentation method provides different regions where there are possible people. The next step is to determine which of those regions effectively contain people and which do not. In order to do that, we apply a semi-elliptical contour model, similar to the model used in [11]. The semi-elliptical contour model consists of two semi-ellipses describing the torso and the human head. The contour model is represented with an 8- dimensional state vector:

$$d^t_{body} = (x_T, y_T, w_T, h_T, x_H, y_H, w_H, h_H) \tag{2}$$

where (x_T, y_T) is the mid-point of the ellipse describing the torso with width w_T and height h_T, and (x_H, y_H) is the mid-point of the ellipse describing the head with width w_H and height h_H.

For each region obtained by the segmentation method an elliptic model is fitted setting the mid-point of the torso ellipse to the center of the region. Since people width varies with distance, we determine the dimension of the elliptic model using a table containing the means of the width of torso and head for five different people at different distances to the robot. This constraint avoids considering regions whose dimensions are incomplatible with the dimensions of people at specific distances. At the same time, this constraint enables the robot to achieve a better fit for the semi-ellipsis describing people.

As different people have usually different width, we adjust the contour people model by varying the mid-point of the torso as well as its width and height. To

determine which variation has the best fit, we evaluate the fitting probability as the probability of the parameters of the contour people model given a person p_i, that we denote as $P(d^t_{body}|p_i)$. The probability is calculated as follows:

$$P(d^t_{body}|p_i) = maxValue\frac{L_f(d^t_{body})}{N_T} \tag{3}$$

where L_f is the number of points in the ellipse that fit with an edge of the image given the parameters of the model d^t_{body}, v denotes the different variations of the model and N_T is the total number of points in the contour model. The edges of the image are calculated applying the Canny edge detector [14].

Finally, since the accuracy of the disparity picture decreases with distance, we reformulated Equation (3) as follows:

$$P(d^v_{body}|p_i) = \begin{cases} \dfrac{P(d^v_{body}|p_i)}{R \cdot e^{-\rho\mu_z}} & if \;\; \mu_z \leq \delta \\ 0 & if \;\; \mu_z > \delta \end{cases} \tag{4}$$

where R is a normalization parameter, ρ denotes the resolution at which the distance can be measured and δ is the maximum distance allowed. This probability decreases with the distance Z represented by the average μ_z of the distances of all points belonging to region i.

5 Skin Detection

In addition to the elliptical people model, we employed a probabilistic model for skin detection. Skin detection is the process of finding pixels that represent color skin in an image or a video labeling the pixels whether they represent skin or a non-skin pixel. The skin is detected using a color image of the stereo camera and applying the method proposed by Gómez et al. [10]. This method consists on a combination of components of different color spaces. A color space is a model that refers to the color composition in terms of variables such as intensity, luminance, brightness, etc. Some of the most commonly used color models are RGB, HSV, YCrCb, YES, and RGBy. The combination of color spaces obtained by Gomez et al. [10] allows to detect skin in images captured in both indoors and outdoors environments. To select the most appropriate components for detecting skin they used two types of images (i) skin and non-skin in indoor environments, and (ii) skin and non-skin in outdoors environments. The two sets of images covering more than 2000 people of different races and ages. These sets of images were used to obtain 3.350 regions containing skin information. The average size of the regions were 35×35 pixels. By using machine learning tools such as CN2 [13] the following formula was obtained to detect pixels with skin information in an image:

$$SkinDetected = true \; if \; (E > 13.4224) \, and \, ((red/green) < 1.7602) \tag{5}$$

where the component E is calculated as $E = \frac{1}{2}(red - green)$.

The skin detection process using Equation (5) is performed independently in each pixel of the image. This type of detection technique is called pixel-based skin detection, which has the characteristic of not assuming any dependency between the pixels of the image.

5.1 Probabilistic Skin Model

The skin color model determines whether a pixel in the image represents a skin or not. However, it is common to detect false positives in certain regions of the image. Moreover, it is desirable to assign a skin detection probability rather than the true/false output of Equation (5). Therefore, we use a Skin Probability Map (SPM) [12] for skin detection. An SPM is a histogram in which the occurrences of each possible value of the components of a color model are stored. We use two SPMs, one for each set of training images, skin and non-skin. The pixels of each image of the training set were labeled as skin or not using the equation 5. Since the training sets of images were taken from environments similar to those in which the robot is expected to execute their tasks, the color histograms obtained can be considered as probability distribution functions for the detection of skin and non-skin in such environments. This allows not only to obtain a probability for skin detection but also to eliminate false positives in the detection. The RGB color model was used to obtain the histograms for skin $Hist_{skin}$ and non-skin $Hist_{non-skin}$. Thus, if each element of the color histogram is divided by the total number of points it is possible to obtain a probability of skin for a given element [r, g, b]. The conditional probability of a pixel given that it represents skin or non-skin is determined by applying Equations (6) and (7):

$$P(rgb_{(x,y)}|skin) = \frac{Hist_{skin}[r,g,b]}{Total_{skin}} \tag{6}$$

$$P(rgb_{(x,y)}|non-skin) = \frac{Hist_{non-skin}[r,g,b]}{Total_{non-skin}} \tag{7}$$

where $Total_{skin}$ is the total number of elements in the histogram of skin $Hist_{skin}$ and $Total_{non-skin}$ is the total number of elements in the histogram of non-skin $Hist_{non-skin}$.

To obtain the probability of skin given a person p_i, the probabilities of skin on each pixel are averaged according to the image region i to which they belong as follows:

$$P(skin|p_i) = \frac{\sum_{x=0}^{M}\sum_{y=0}^{N} P(rgb_{(x,y)}|skin)}{N \times M} \tag{8}$$

where $M \times N$ is the size of the region containing the person p_i, and $rgb_{(x,y)}$ is the RGB value for the pixel (x, y).

To calculate $P(skin|p_i)$ we only consider the pixels that were detected with the SPM as skin using Equation (9):

$$\frac{P(rgb_{(x,y)}|skin)}{P(rgb_{(x,y)}|non-skin)} \geq \theta \tag{9}$$

where θ is obtained experimentally as 0.8 and represents the threshold at which can be determined the presence of skin.

In a similar way to the elliptical model, the distance at which the region is detected is considered in the skin model since the accuracy of the disparity image decreases with distance. Therefore, the final expression for calculating the probability of people detection based on our skin model is defined as follows:

$$P(skin|p_i) = \begin{cases} \frac{P(skin|p_i)}{R \cdot e^{-\rho\mu_z}} & if \;\; \mu_z \leq \delta \\ 0 & if \;\; \mu_z > \delta \end{cases} \tag{10}$$

where R is a normalization parameter, ρ denotes the resolution at which the distance can be measure and δ is the maximum distance allowed. This probability decreases with the distance Z represented by the average μ_z of the distances of all points belonging to region i.

6 People Detection and Tracking

At this point, we can determine with some probability, the presence of people in an image using the contour of people and skin evidence independently. The integration of the evidence was performed by a weighted sum of their probabilities using Equation (11):

$$P(skin, contour|p_i) = \alpha \times P(skin|p_i) + (1-\alpha) \times P(d^v_{body}|p_i) \tag{11}$$

where α is a weight associated with evidence of skin and silhouette which is experimentally determined as 0.4.

In order to improve the detection process we use video streaming and combine evidence of several frames with a Bayesian scheme. Once a person has been detected at time t , we proceed to search if that person was previously detected, with our proposed method, at time $t-1$ calculating the Euclidean distance from current regions that contains a person to previous regions. If the distance between two regions is less than a threshold value, then we consider these people to be the same. To calculate the probability of a person p_i at time t we apply

$$P(p_i^t) = \frac{P(skin, contour|p_i^t)P(p_i^{t-1})}{P(skin, contour|p_i^t)P(p_i^{t-1}) + P(skin, contour| \sim p_i^t)P(\sim p_i^{t-1})} \tag{12}$$

where $P(skin, contour|p_i^t)$ is the probability of skin and contour given the person p_i at time t which is calculated applying equation 11. $P(p_i^{t-1})$ is the probability of the person p_i at time $t-1$. $P(skin, contour| \sim p_i^t)$ is the probability of skin and contour given that the person p_i was not detected at time t and $P(\sim p_i^{t-1})$ is the probability of that the person p_i has not been detected at time $t-1$. In the experiments the values of $P(p_i^t)$ when $t=0$ were set to 0.5.

7 Experimental Results

We present experimental results that demonstrate how the proposed method, using an ActivMedia PeopleBot mobile robot equipped with a stereo camera, is able to detect people who is standing, sitting and leaning sideways at 12 f.p.s. using both real-time and pre-recorded video. Moreover, it can also detect people from their backs and not facing the robot. In order to test and evaluate our method we use a mobile robot in dynamic indoor environments with different lighting conditions and without assuming a prior visual model of the environment. People was placed at different distances from the robot (1 to 5 m). The number of people varies from one to three.

Figure 2 shows how our people detection method is able to detect multiple people and track them over time. The experiments were performed with a maximum of three people; however, in Figure 2 we show an experiment with two people to compare the performance of the people detection method using a single frame detection scheme against our people detection method using evidence from several frames applying a Bayesian scheme. Snapshots of the experiments with three people are shown in Figure 4. We calculated the detection rate DR as follows:

$$DR = \frac{N_D}{N_T} \tag{13}$$

where N_D is the number of frames where people were detected with $P(p_i) > 0.8$, and N_T is the total number of frames considered.

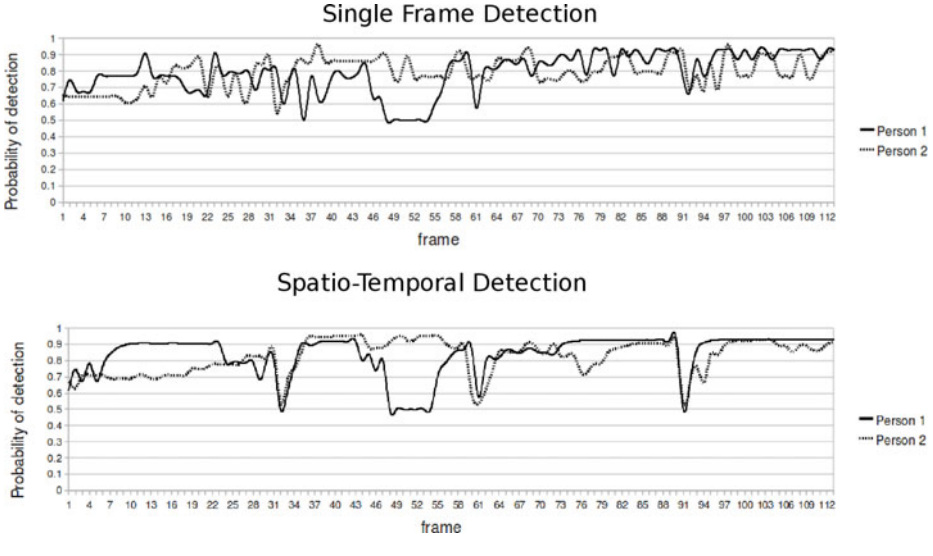


Fig. 2. Multiple people detection performance using single and spatio-termporal detection

Our segmentation method has a DR of 94.0% with one false positive every 1800 regions examined.

Table 1 presents the detection rates of our method merging the evidence of silhouette and skin. A closely related work is proposed in [11], but since they use a thermal camera for people detection, a direct comparison is not possible (their detection rates presented in [11] are in the order of 91.9). Therefore Table 1 only shows a comparison with the results presented in [2]. The detection rate using both skin and contour evidence is lower than the detection rate obtained by using only contour evidence. However, as can be seen in Figure 3, by using only the contour people model we can obtain false positives on the background (Figure 3 (a) and (c)) whereas with our proposed method, fusing the evidences of contour and skin, those false-positives can be successfully filtered.

In Figure 4 one can see the results of different experiments on detection of standing and sitting people, as well as people in frontal and side view with a mobile robot which is also moving in real world scenarios using our proposed people detection method.

Table 1. Detection Rates and False Positives

Method	Detection rate	False Positives / regions examined
Adaptive countour model [2]	89.0	53 / 1800
Adaptive spatial-temporal model with contour model [2]	96.0	32 / 1800
Our method (Merging contour and skin)	94.0	1 / 1800

(a) (b) (c) (d)

Fig. 3. False positives reduction

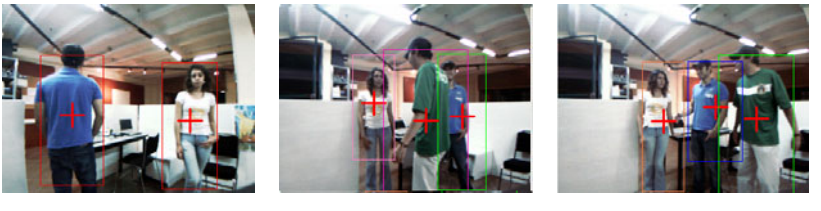


Fig. 4. Experiments performed using a mobile robot in dynamic indoor environments

8 Conclusion and Future Work

This paper focuses on detecting multiple people with a mobile robot by fusing information from different sensors over time and applying a Bayesian scheme without a prior visual information about the environment. The proposed segmentation method takes into account the distance between the robot and people to perform a segmentation and to provide a first estimation of people location. We use an adaptive contour people model based on people distance and a color model of skin to calculate a probabiliy of people etection, one for each evidence. To detect people, we merged the probabilities of the fused model over time by applying a Bayesian scheme. The experimental results show how our proposed method is able to segment and detect standing and sitting people, as well as people in frontal and side view with a mobile robot without assuming a previous visual model of the environment or that people are facing or not the robot. By using only the contour people model we can obtain false positives on the background, whereas, with our proposed method, fusing the evidences of contour and skin, those false-positives can be successfully filtered. As future research work we are considering not only detect people but also recognize them.

References

1. Burgard, W., Cremers, A., Fox, D., Hhnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S.: Experiences with an Interactive Museum Tour-guide Robot. Artificial Intelligence, 3–55 (1999)
2. Méndez-Polanco, J.A., Muñoz-Meléndez, A., Morales, E.F.: People Detection by a Mobile Robot Using Stereo Vision in Dynamic Indoor Environments. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, p. 349. Springer, Heidelberg (2005)
3. Osada, J., Ohnaka, S., Sato, M.: The Scenario and Design Process of Childcare Robot, PaPeRo, Dynamic Indoor Environments, pp. 80–86 (2006)
4. Malagón-Borja, L., Fuentes, O.: Object Detection Using Image Reconstruction with PCA. Image Vision Computing 27(1-2), 2–9 (2009)
5. Cielniak, G., Duckett, T.: People Recognition by Mobile Robots. Journal of Intelligent and Fuzzy Systems: Applications in Engineering and Technology 15(1), 21–27 (2004)
6. Müller, S., Schaffernicht, E., Scheidig, A., Hans-Joachim, B., Gross-Horst, M.: Are You Still Following Me? In: Proccedings of the 3rd European Conference on Mobile Robots ECMR, Germany, pp. 211–216 (2007)
7. Schaffernicht, E., Martin, C., Scheidig, A., Gross, H.-M.: A Probabilistic Multimodal Sensor Aggregation Scheme Applied for a Mobile Robot. In: Proccedings of the 28th German Conference on Artificial Intelligence, Koblenz, Germany, pp. 320–334 (2005)
8. Lastra, A., Pretto, A., Tonello, S., Menegatti, E.: Robust Color-Based Skin Detection for an Interactive Robot. In: Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence (AI*IA 2007), Roma, Italy, pp. 507–518 (2007)
9. Han, J., Bhanu, B.: Fusion of Color and Infrared Video for Dynamic Indoor Environmentsving Human Detection. Pattern Recognition 40(6), 1771–1784 (2007)

10. Gómez, G., Sánchez, M., Sucar, L.E.: On Selecting an Appropriate Colour Space for Skin Detection. In: Coello Coello, C.A., de Albornoz, Á., Sucar, L.E., Battistutti, O.C. (eds.) MICAI 2002. LNCS (LNAI), vol. 2313, pp. 69–78. Springer, Heidelberg (2002)
11. Treptow, A., Cielniak, G., Ducket, T.: Active People Recognition using Thermal and Grey Images on a Mobile Security Robot. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Edmonton, Alberta, Canada (2005)
12. Jones Michael, J., Rehg James, M.: Statistical Color Models with Application to Skin Detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1999), vol. 1, p. 1274–1280 (1999)
13. Clark, P., Boswell, R.: Rule induction with CN2. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS (LNAI), vol. 482, pp. 151–163. Springer, Heidelberg (1991)
14. Canny, J.: A Computational Approach To Edge Detection. IEEE Transactions Pattern Analysis and Machine Intelligence (1986)

A Motion Planner for Finding an Object in 3D Environments with a Mobile Manipulator Robot Equipped with a Limited Sensor

Judith Espinoza and Rafael Murrieta-Cid

Centro de Investigación en Matemáticas, CIMAT
Guanajuato, México
{jespinoza,murrieta}@cimat.mx

Abstract. In this paper, we address the problem of searching an object with a mobile robot in a known 3-D environment. We consider a 7 degrees of freedom mobile manipulator with an "eye-in-hand" sensor. The sensor is limited in both range and field of view. In this work, we *propose a solution to the "where to look" part of the object finding problem* based on three main procedures: 1) We propose a practical and fast method to approximate the visibility region in 3D of a sensor limited in both range and field of view. 2) We generate candidate sensing configurations over the robot configuration space using sampling. 3) We determine an order for visiting sensing configurations using a heuristic.

We have implemented all our algorithms, and we present simulation results in challenging environments.

1 Introduction

In this paper, we address the problem of searching a static object with a mobile robot in a known 3-D environment. We consider a 7 degrees of freedom mobile manipulator with an "eye-in-hand" sensor. The sensor is limited in both range and field of view. Fig. 1 (a) shows this robot and the visibility region of the limited sensor, which has the shape of a frustum.

Furthermore, our planner aims to diminishing the expected value of the time for finding the object. This could be very useful in applications in which the time assigned to the task is limited or not completely known. We believe that our method has a wide range of applications, from finding a specific piece of art in a museum to search and detection of injured people inside a building.

In [2,5] it has been shown that the problem of determining the global order for visiting sensing locations, which minimizes the expected value of the time to find an object is NP-hard, even in a 2-D polygonal workspace with a point robot. The present work addresses a related problem but in a 3D workspace with a mobile manipulator robot equipped with a limited sensor. Hence, due to the computational complexity of the tasks we have to solve and integrate in a motion strategy, we propose approximated solutions based on sampling and heuristics. In [4] an approach has been proposed for finding an object with a

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 532–541, 2010.

mobile manipulator, but in [4] the sensor used by the robot was omnidirectional (i.e. it has a field of view of 360 deg.) and without range limit. Furthermore, the environments considered in [4] were small and geometrically simple. In this paper we propose a motion planner for the more challenging case in which the robot is equipped with a sensor limited in both range and field of view, and the robot is moving in large and geometrically complex 3D environments (see Fig 2).

Our search problem is related to art gallery problems and coverage. The traditional art gallery problem is to find a minimal placement of guards such that their respective visibility regions completely cover a polygon [10].

In coverage problems (e.g., [7,1]), the goal is usually to sweep a known environment with the robot or with the viewing region of a sensor. In this problem, it is often desirable to minimize sensing overlap so as not to cover the same region more than once. Our problem is related to the coverage problem in the sense that any complete strategy to find an object must sense the whole environment.

In this work, we *propose a solution to the "where to look" part of the object finding problem* based on three main procedures: 1) We propose a practical and fast method to approximate the visibility region in 3D of a sensor limited in both range and field of view. 2) We generate candidate sensing configurations over the robot configuration space also using sampling. We select a set of sensing configurations having two properties: a low cardinality of sensing configurations and a small cost to move between them. Both properties are convenient for diminishing the average time to find the searched object. 3) We determine an order for visiting sensing configurations using a heuristic.

We use numerical optimization methods for diminishing the cost of moving the robot during the object finding task. To reduce the computational running time of this procedure, we select the most important robot degrees of freedom (DOFs) and carry out the optimization procedure only considering this reduced number of DOFs.

2 General Approach and Notation

In our object finding problem, the expected value of the time to find an object depends on two main factors: 1) the *cost* of moving the robot between two configurations, quantified in a given metric, and 2) the probability mass of seeing the object, which is equivalent to the *gain*.

In our formulation, we assume that the environment is known, but that we do not have any information about the location of the object being searched. Since there is no reason to believe that one object location is more likely than another, we assign equal values to every location. This is equivalent to defining an uniform probability density function (pdf) modeling the object location. Indeed, this assumption is already known in the literature and it is called the principle of insufficient reason [6]. We believe this reasoning is general given that we do not need to assume a relation between a particular type (class) of object and its possible location (for instance, balloons are floating but shoes lie on the ground), which could reduce the scope of the applications.

The robot senses the environment at discrete configurations q_i (also known as *guards*, from the art gallery problem [10]). Let us call $V(q_i)$ the visibility region associated to the limited sensor. Since the pdf modeling the object location is uniform, the probability of the object being in any subset $V(q_i) \subseteq int(W)$ is proportional to the volume of $V(q_i)$. The interior of the workspace $int(W)$ is the free space inside the 3D environment.

To compute the probability mass of seeing the object in a 3D environment, –gain, we need to calculate the size and shape of visibility regions and their intersections, but the computational complexity of such a task in a 3D environment populated with obstacles is very high [8]. For this reason, we decided to use a sampling scheme to calculate an approximated visibility region of the limited sensor.

Our searching strategy is as follows: first the whole environment is divided into a set of convex regions. To divide the environment into convex regions we use the probabilistic convex cover proposed in [3][1]. This method divides the environment into a set called $\{C(r)\}$, so that the union of all $C(r)$ covers the whole environment, that is $\bigcup_r C(r) = int(W)$. $C(r)$ denotes a convex region in a 3-D environment, and r indexes the region label. Note that all points inside $C(r)$ can be connected by a clear line of sight from any point $p(x, y, z)$ inside $C(r)$.

Second, each convex region is covered with the sensor frustum denoted by $\mathcal{F}$. We establish a route to cover the whole environment by decomposing the problem into two parts: First an order to visit convex regions $C(r)$ is established. Second, sensing configurations in a configuration space $\mathcal{C}$ of 7 dimensions are generated to collectively cover each convex region, the sensing configuration are linked in a graph and perform a graph search to generate the search trajectory.

We believe that there exists an analogy between our searching protocol and painting a house, first the order for painting rooms (equivalent to the order for visiting the convex regions) in the house is established. Then every wall in each room is painted, this is equivalent to visit our sensing configurations, whose associated view frustums cover each single convex region.

Since the expected value of the time depends also on the cost of moving the robot between sensing configurations, we need to find shortest paths to move the robot. The actual paths depend on the metric used to measure cost. One way to define the cost between two configurations X and Y in a D-dimensional configuration space is

$$\|X - Y\|_\Lambda \equiv (X - Y)^T \Lambda (X - Y), \tag{1}$$

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 joints might be not all equivalent (e.g., large links vs. small links), and also because different DOFs might not even be measuring the same movement (translation vs. rotation).

[1] Notice that in [3] the environment is 3D, but the robot is a point equipped with an omnidirectional sensor without range limit.

To find the shortest path between two convex regions we use the wavefront expansion (called NF1) proposed in [9].

The two orders: (1) for visiting convex regions, and (2) for visiting sensing configurations inside a single convex region are established based on the utility function proposed in [5]. This utility function measures how convenient it is to visit a determined configuration from another, and is defined as follows:

$$U\left(q_k, q_j\right) = \frac{P\left(q_j\right)}{Time\left(q_k, q_j\right)}. \tag{2}$$

This means that if a robot is currently in q_k, the utility of going to configuration q_j is directly proportional to the probability of finding the object there and inversely proportional to the time it must invest in traveling. A robot using this function to determine its next destination will tend to prefer configurations that are close and/or configurations where the probability of seeing the object is high.

The utility function in (2) is sufficient to define a 1-step greedy algorithm. At each step, simply evaluate the utility function for all available configurations and choose the one with the highest value. This algorithm has a running time of $O\left(n^2\right)$, for n configurations.

However, it might be convenient to explore several steps ahead instead of just one to try to "escape local minima" and improve the quality of the solution found. So, we use this utility function to drive the algorithm proposed in [5] to search a reduced space. This algorithm is able to explore several steps ahead without incurring a too high computational cost. This algorithm is described in detail in [5].

3 Selecting Sensing Configurations

The method that we propose to cover each convex region is based on sampling. There have been many approaches that use samples to capture connectivity of high dimensional spaces, for example, [13], [12] just to name a pair. Notice that in our work we also want to connect sensing configurations, but we have an additional goal. *We are interested in representing the free space inside the workspace for searching an object*, and not only for abstracting the configuration space for path planning purposes.

Sensing configurations $q_{(i,r)}$ are generated with a uniform probability distribution in a configuration space $\mathcal{C}$ over 7 dimensions: Two coordinates (x, y) defining the position of the robot's base and five angles. One angle measures the orientation of the robot's base, and the other 4 measure the orientations of the arm's links. A sensing configuration $q_{(i,r)}$ is associated to a given region $C(r)$. Each convex region has associated a set $S(r)$ of point samples $s(r) \in S(r)$ originally used to compute the size and shape of a convex region (see [3]). Each point sample $s(r)$ lies in the 3D space, and is defined by a 3-dimensional vector $p(x, y, z)$. We use this previously computed set of points to determine the coverage of a convex region with a limited sensor.

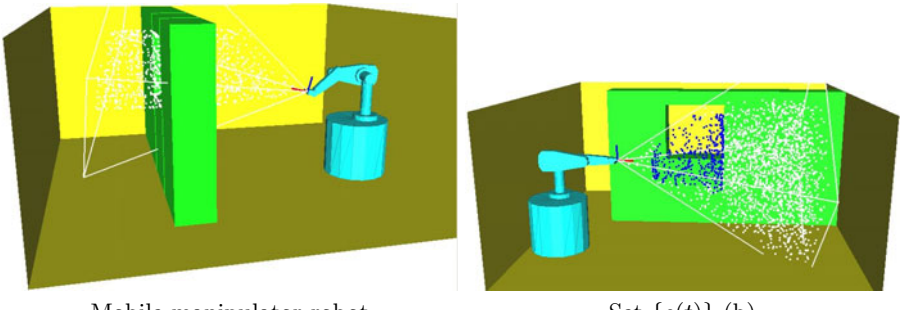
Notice that *a set of sensing configurations with minimal cardinality might not be the optimal one for our problem*, since the cost for reaching the configurations might be large if they are far from one another. Thus, our problem is different from the problem of only choosing the minimum number of configuration that collectively sense the whole environment: The art gallery problem. Since, even the easier art gallery problem is NP-hard [11], it is unlikely that our problem can be solved by a deterministic algorithm efficiently. For this reason, we decided to use a sampling scheme to compute $\{q_{(i,r)}\}$.

Our algorithm for selecting sensing configurations has been inspired from the algorithm presented in [14]. That method is designed to cover the boundary of a polygonal –2D– workspace ∂W. In our problem, we aim to cover the free space inside the polyhedral –3D– environment representation.

In our method, the point samples lying inside the frustum associated to a sensing configuration $q_{(i,r)}$ are used to approximate the actual visibility region $V(q_{(i,r)})$. The robot's configurations used to cover a convex region have the property that all of them place the sensor inside the convex region being sensed. This property allows us to approximate the visibility region of the limited sensor without complex 3D visibility computations. The visibility region of the limited sensor at configuration $q_{(i,r)}$ is approximated by:

$$V(q_{(i,r)}) = \bigcup_{s} s(r) \in int(\mathcal{F} \cap C(r)) \tag{3}$$

In figure 1 (a), the white dots are used to show the point samples $s(r) \in S(r)$ covered by the view frustum $\mathcal{F}$ and inside a convex region $C(r)$.



Mobile manipulator robot and its visibility region (a)

Set $\{s(t)\}$ (b)

Fig. 1. Visibility Region and Set $\{s(t)\}$

Notice that the sensor can be inside several convex regions at once, in such a case, the visibility region is approximated as the point samples lying simultaneously inside the frustum and the convex regions having the sensor in its interior.

While covering region $C(r)$, we also mark as sensed and logically remove, all samples $s(t)$ belonging to region $C(t), t \neq r$, if $s(t) \in int(\mathcal{F} \cap C(r) \cap C(t))$. It is guaranteed, that these samples are not occluded from configuration $V(q_{(i,r)})$.

In figure 1 (b) dark (blue) dots are used to show the set $S(t)$, and white dots represent the set of point samples $S(r)$ belonging to the region in which the sensor resides and inside the frustum.

A convex region $C(r)$ is totally covered if:

$$\bigcup_{s} s(r) \in int(C(r)) = S(r) \tag{4}$$

Similar to [14], we select sensing configurations based on the cardinality of its point samples. Iteratively, we select the configurations with the largest cardinality of point samples $s(r)$ until all the set $S(r)$ is sensed. The point samples associated to selected sensing configurations are logically *removed*. Thus, redundant sensing configurations, with low point samples cardinality are avoided, yielding a reduced set containing only sensing configurations with high cardinality of point samples and a small number of redundant point samples.

Additionally, in our sensing configuration sampling scheme, we reject candidate sensing configurations in whose view frustum is in collision with the robot itself, thus, avoiding occlusions generated by the robot body. We also reject sensing configurations, that produce a collision of the robot with the obstacles and robot self-collisions.

The procedure of selection of sensing configurations described above is repeated several times for each convex region $C(r)$, yielding different sets $\{q_{(i,r)}\}$ of sensing configurations. We select one of those sets based on the average value of its graph edges. We describe this last selection below.

4 Connecting Sensing Configurations

Since we want to have options to move the robot between sensing configurations, and thus further reduce the expected value of the time to find the object, we connect the sensing configuration of each set $\{q_{(i,r)}\}$ into a fully connected graph with ne edges. We assign weights $w_{(i,j)}$ to the graph edges according on the metric used to measure the cost of moving the robot between sensing configurations q_i and q_j.

We select the set that minimizes the average value of the graph edges, that is $\min(\frac{1}{ne}\sum_{<i,j>} w_{(i,j)})$. At the end, we obtain a set having both: low cardinality of sensing configurations and small cost to move between sensing configurations.

To cover a region with a limited sensor several sensing configuration are required. To reduce the running time of planning robot paths to cover a region with a limited sensor, we consider that the cost of moving the robot between sensing configurations corresponds to a straight line in the configuration space. That is, for reducing the computational time to cover the environment with a limited sensor, we *pre-estimate* the cost to move between sensing configuration as a straight line in the configuration space.

In the motion planning problem of generating collision free paths to move between configurations, we use a lazy collision checking scheme [15]. The sensing configurations are connected in the graph without checking whether or not a collision free path can connect them. Since we proceed visiting convex regions one by one, it is likely to find collision free paths among configurations to cover the same convex region. Often a small region can be covered with small robot motions, and big regions offer large open space to move the robot. We postpone the collision checking until an order of sensing configurations is established. If some edge of the route is in collision then a new sub-order is generated using other edges in the graph. The new sub-route starts from the last configuration that was connected in the route, and includes all the remaining sensing configurations. Evidently, sometimes the fully connected graph splits into two connected components, if so, we use an RRT [16] to find a collision-free path between the two components. In this later case, the cost assigned to the graph edge connecting the two associated sensing configurations is the cost of the path, which has been obtained with the RRT and checked for collision. In our experiments, we have found that this strategy significantly reduces the time to generate collision-free paths to cover convex regions.

Note that we use the convex regions to facilitate the object finding task. There are four main reasons to do so: First, we avoid the use of complex data structures to update the portion of the environment that has been covered. A convex region is covered if the samples used to approximate its volume have been sensed, or equivalently if the union of the frustums associated to sensing configurations has collectively covered all the point samples inside a convex region. Second, we take advantage of our convex covering to postpone the collision checking until an order of sensing configurations is established. To cover a convex region, it is often possible to find a collision free path between sensing configurations by simply moving the robot in a straight line path in the configuration space. Third, the robot does not move to cover another convex region until a given convex region has been totally covered. It avoids unnecessary robot motion to visit sensing configurations far away from a previous one. Fourth, once a global plan is generated if the environment changes locally then the global plan can also be modified *locally*, only considering the convex regions related to the change in the map.

4.1 Optimizing a Reduced Number of DOFs

The order of visiting regions is established based on a path of optimal cost between regions. This optimal path is computed with the wavefront expansion algorithm described in [9]. Regardless the optimization algorithm used, the problem of finding the path of locally optimal cost (between two robot's configurations) for a robot with several DOFs can take significant amount of time, specially for a global path including a large number of inter-medial sensing configurations.

Indeed, in our experiments we have found that bottleneck limitation regarding the running time of our algorithms corresponds to compute the optimal paths for a large number (4 or more) of DOFs. To mitigate this problem, we select some

DOFs and carry out the optimization procedure over this reduced number. We determine the other DOFs using a randomized sampling procedure. Thus, a robot path to move between convex regions is a sequence of robot's configurations, in which some DOFs are planned optimally and the other do not produce collisions between the robot and the obstacles.

In terms of energy and time is more costly to translate a mobile manipulator robot than rotate its base or its arm links. In fact, if the planner is able to coordinate the translation of the robot base and the rotations of the arm's links, such that both translation and rotations happen at once, then the cost of moving the arm is zero (in terms of elapsed time), since the motions (translation and rotations) are simultaneous. Furthermore the environment in which the robot resides is large, and the robot has to translate a long distance (equivalent to time when the robot moves with saturated speed) to search the object. Consequently, we consider that the coordinates (x, y) defining the position of the robot's base are the most important DOFs for this problem. Hence, we optimize only these two DOFs.

5 Simulation Results

All the results presented in this paper were obtained with a regular PC running Linux OS, equipped with an Intel Pentium IV microprocessor and 1 Gigabyte of RAM. Figures from 2 (a) to 2 (d) show snapshots of the object finding task in an indoor environment. This environment cannot be searched by using a 2D projection; since information about the 3D structure will be lost. Notice the stairs, the windows, the lamp and the furniture.

For this environment, our convex cover algorithm, produced 47 convex regions. This convex cover decomposition was computed by our software in 2 minutes and 40 seconds. 552 sensing configurations were needed to cover the whole environment with the limited sensor. The running time of our software to generate these 552 sensing configurations, link them in a graph in C, and computing an order to visit them was 1 hour and 33 minutes.

Figure 2 (a) shows with two (blue) meshes 2 convex regions, called respectively region $C(A)$ and $C(B)$. Convex regions $C(A)$ and $C(B)$ are consecutive in the order to visit convex regions. Figure 2 (a) also shows the path to move between regions $C(A)$ and $C(B)$.

Figure 2 (b) shows the 19 sensing configurations needed to collectively cover region $C(A)$. Figure 2 (c) shows the path to move between two sensing configurations associated to region $C(A)$, this path corresponds to a straight line in a configuration space C of 7 dimensions.

Only $\frac{1}{10}$ of the total number of paths to sense the whole map was computed with a RRT. All the other times, a straight line in C was enough to find collision free paths.

Figure 2 (d) shows the global robot's path to visit all the convex regions. In this global path only the DOFs (x, y) defining the robot's base position are optimized. The time to compute this global path was 15 minutes and 9 seconds.

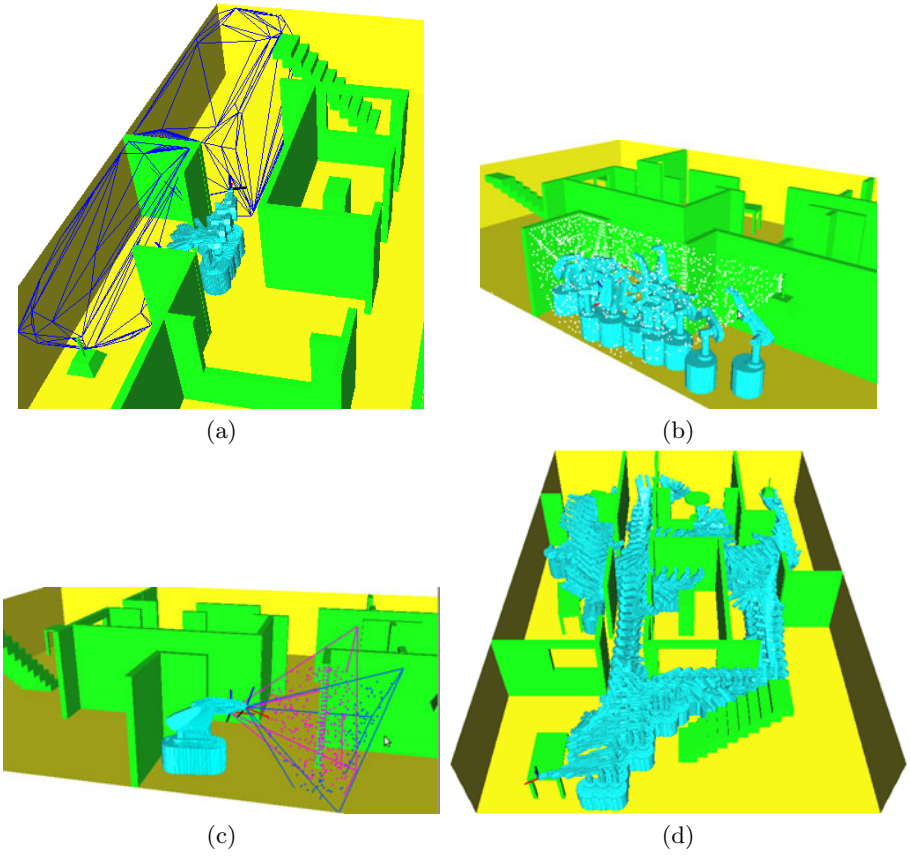


Fig. 2. Finding an object with a limited sensor

The total running time of our software to compute a plan for sensing the whole environment with a limited sensor was 1 hour and 50 minutes. The expected value of the time is 100.44 units.

To our knowledge, this is the first method able to solve this problem with a 7 DOFs robot equipped with a limited sensor in a realistic 3D map.

6 Conclusion

In this paper, we have proposed a motion planner for finding an object in 3-D environments with a robot equipped with a limited sensor. Our main contributions are: (1) We have proposed a practical and fast method to approximate the visibility region in 3D of a sensor limited in both range and field of view. (2) We have proposed a method to select candidate sensing configurations having convenient properties for diminishing the average time to find the searched object. (3) We have proposed the strategy of selecting the most important DOFs

to be optimized. This strategy significantly reduces the computational running time to find the object. We have implemented all our algorithms, and we have presented simulation results in challenging environments.

References

1. Acar, E.U., Choset, H., Atkar, P.N.: Complete sensor-based coverage with extended-range detectors: A hierarchical decomposition in terms of critical points and voronoi diagrams. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IEEE/RSJ-IROS (2001)
2. Sarmiento, A., Murrieta-Cid, R., Hutchinson, S.: A Multi-robot Strategy for Rapidly Searching a Polygonal Environment. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) IBERAMIA 2004. LNCS (LNAI), vol. 3315, pp. 484–493. Springer, Heidelberg (2004)
3. 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, IEEE-ICRA 2005 (2005)
4. Sarmiento, A., León-Espinoza, J., Murrieta-Cid, R., Hutchinson, S.: A Motion Planning Strategy for Rapidly Finding an Object with a Mobile Manipulator in 3-D Environments. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008. LNCS (LNAI), vol. 5317, pp. 562–572. Springer, Heidelberg (2008)
5. Sarmiento, A., Murrieta, R., Hutchinson, S.A.: An Efficient Motion Strategy to Compute Expected-Time Locally Optimal Continuous Search Paths in Known Environments. Advanced Robotics 23(12-13), 1533–1569 (2009)
6. LaValle, S.: Planning Algorithms. Cambridge University press, Cambridge (2006)
7. Hert, S., Tiwari, S., Lumelsky, V.: A terrain-covering algorithm for an auv. Autonomous Robots 3, 91–119 (1996)
8. Goodman, J.E., O'Rourke, J. (eds.): Handbook of Discrete and Computational Geometry. CRC Press, Boca Raton (1997)
9. Latombe, J.-C.: Robot Motion Planning. Kluwer Academic Publishers, Dordrecht (1991)
10. O'Rourke, J.: Art Gallery Theorems and Algorithms. Oxford University Press, Oxford (1987)
11. Shemer, T.: Recent Results in Art Galleries. Proc. IEEE 80(9), 1384–1399 (1992)
12. Hsu, D., Latombe, J.C., Motwani, R.: Path planning in expansive configuration spaces. In: Proc. IEEE Int. Conf. on Robotics and Automation, IEEE-ICRA 1997 (1997)
13. 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)
14. González, H.H., Latombe, J.-C.: A Randomized Art-Gallery Algorithm for Sensor Placement. Proc. 17th ACM Symp. on Computational Geometry, SoCG 2001 (2001)
15. Sanchez, G., Latombe, J.C.: A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking. In: Jarvis, R.A., Zelinsky, A. (eds.) ISRR 2001. STAR (2003)
16. LaValle, S.M., Kuffner, J.J.: Randomized kinodynamic planning. International Journal of Robotics Research 20(5), 378–400 (2001)

A Driving Assistance System for Navigation in Urban Environments

Leandro C. Fernandes, Maurício A. Dias, Fernando S. Osório, and Denis F. Wolf

USP – University of São Paulo / ICMC – SSC
LRM – Mobile Robotics Laboratory
São Carlos, SP - Brazil
{lnd,fosorio,denis}@icmc.usp.br,
macccdias@gmail.com

Abstract. Mobile robot navigation in urban environments is a very complex task. As no single sensor is capable to deal with these situations by itself, sensor fusion techniques are required to allow safe navigation in this type of environment. This paper proposes an approach to combine different sensors in order to assist a driver in a cooperative manner. An adaptive attention zone in front of the vehicle is defined and the driver is notified about obstacles presence, identifying dangerous situations. Experiments using a commercial vehicle loaded with GPS and a LIDAR sensor have been performed in real environments in order to evaluate proposed approach.

Keywords: Mobile Robot Navigation, Urban Environment, Driver Assistance, Sensor Fusion, Vehicles Safety Systems.

1 Introduction

Mobile robotics is a multidisciplinary research area that aims to develop machines capable to use sensors to perceive the environment, make decisions, and move autonomously to complete assigned tasks. Although most research work in this field deals with autonomous systems, there are several other applications for the perception and decision making algorithms used in robotics.

Autonomous vehicles development has been receiving considerable attention by robotics community in the last five years. Initiatives like DARPA Grand Challenge [1], DARPA Urban Challenge [2] and ELROB [3] have been concentrating efforts of several universities and research institutes to push the state of the art in outdoor navigation algorithms. Although autonomous cars are not yet available commercially, several techniques developed by robotics researchers can already be applied in day by day problems.

Car accidents are one of the major causes of death in the world. Traffic accidents are already the major cause of death of young people in U.S. According NHTSA (US National Center for Statistics and Analysis) [4]. In United States (2002) occurred 6.3 million cars crashes, 2.9 million people got hurt and 42,600 people died. In Brazil the number car crashes are up to 1.5 million with 35 thousands deaths per years. In large cities, like São Paulo, statistics demonstrate that more peoples die by urban traffic crashes than by attacks or natural causes [5].

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 542–551, 2010.

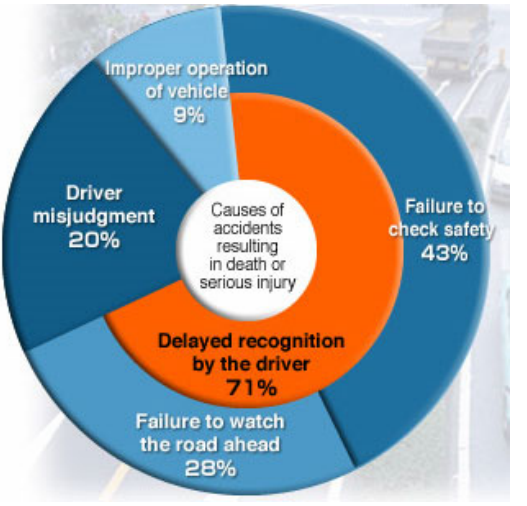


Fig. 1. Causes of accidents resulting in deaths or serious injury [6]

Based on [6], 71% of the car crashes that result in death or severe injury happen due to delayed recognition of the scene by the driver, 43% are due to failure to check safety, and 28% failure to watch the road ahead (Fig 1). Many road accidents could be avoided if the driver were notified of a dangerous situation. In many cases, this task can be performed by sensors that provide information about the environment and a computational system capable of interpreting the sensor data.

Another important application to driving assistance systems is making driving easier to elderly and handicapped people. As the overall population becomes older, the number of older people driving in the streets has increased. A research carry of Johns Hopkins [7] evaluated vision, cognition, and health alterations among 1200 drivers with age into 67 and 87 years. Around 1.5% of them gave up to drive by themselves and more 3.4% reduced intentionally their time at the steering well motivated by visual capacity reduction. In these cases, a system capable of notifying the driver of a possible collision can substantially reduce the risk of a car accident.

Automotive industry has put a considerable effort in the development of new systems to improve safety; however most of these systems do not perform in a proactive manner. Recently, Volvo developed the LKS (Lane Keeping Support), which generates an alarm in case of drowsiness or fatigue of the driver and when a unexpected route changes is detected. This system also assists in a maneuvering by signaling the presence of other vehicles in blind spots. ImapCar [8] is also a commercial systems capable of detecting the presence of people, obstacles and other cars ahead of the vehicle, as well as identifying a traffic lane crossing like. In this case, a video camera is used to acquire information about the surroundings. As the camera cannot estimate the distance to the obstacles, the precision of obstacle detection is very limited.

The techniques developed in the DARPA challenges and EUROB are very efficient for autonomous navigation in urban environments. On the other hand the cost of such systems is well above 1 million dollars, which is prohibitive for most

commercial applications. It is also necessary very accurate GPS information of the traffic lanes, crossroads and other details of the environment to use these techniques appropriately. Other simpler approaches have also been developed like [9], which consists of a low cost detection and obstacle avoidance based on ultra sound sensor system. However only very close obstacles (5 or 6 meters away) are detected.

This work proposes an approach to a driver assistance system based on GPS and LIDAR sensors. The GPS provides global position of the vehicle (allowing for trajectory anticipation), while the LIDAR can accurately detect the presence of obstacles ahead of the vehicle. An adaptive attention zone in front of the vehicle is defined and the driver is notified about presence of obstacles, identifying dangerous situations.

2 DAS Hardware and Software Implementation

The proposed Driving Assistance System (DAS) is composed by a commercial vehicle (Fig. 2) equipped with an integrated hardware and software system capable of detecting and warning the driver about frontal obstacles. The hardware is composed of three different types of sensors (LIDAR, GPS and Compass), a CPU (embedded PC), and a software system. This set work together to sensing the environment, logging and fusion data, generating obstacles detection and to emit collision warnings.

Fig. 2. Vehicle used in experiments equipped with a frontal Sick LMS 200 Laser sensor and with a compass and GPS system

2.1 Sensors Used

A SICK LMS291 [10] is a LIDAR (Light Detection and Ranging device) sensor used to detect obstacles. It has been set with a maximum range of 80m, covering 180

degrees at 37,5Hz. Using an infrared laser beam, this sensor type can detect obstacles accurately even in the complete darkness, compensating some human vision limitations. The LIDAR sensor has been mounted in front bumper of the vehicle where it is capable to detect obstacles ahead of the car. We also had tested other configurations using the sensor on the top of the vehicle, pitching it down to detect bumps and depressions. However, this is beyond the scope of this paper.

A Garmin GPS 18x-5Hz [11] was also part of the experimental setup. Although the GPS provides latitude, longitude, altitude, heading and velocity, the accuracy of this information depends largely of the number of satellites available and proximity between them (Dilution of Precision). Usually, the horizontal error in the position estimation ranges from 3 to 10m in a clear sky situation. In order to increase the heading precision from GPS receiver, a TNTC Revolution [12] digital compass has been used.

2.2 System Implementation

The implemented system uses the data acquired by the LIDAR sensor and fusing with GPS and Compass data to establish a differentiated attention zone, as demonstrated in Fig. 3. One important particularity of this system is the fact that GPS data have been previously collected in the very same path we are expecting to follow using the vehicle.

Initially the distances from the vehicle to any obstacles in a range of 180 degrees with a radius of 80 meters are measured, resulting into a vector of 361 distance measurements readings (0^o to 180^o with step of 0.5^o). Then, the GPS position and compass orientation is obtained in order to determine the vehicle positioning related to a previously defined path.

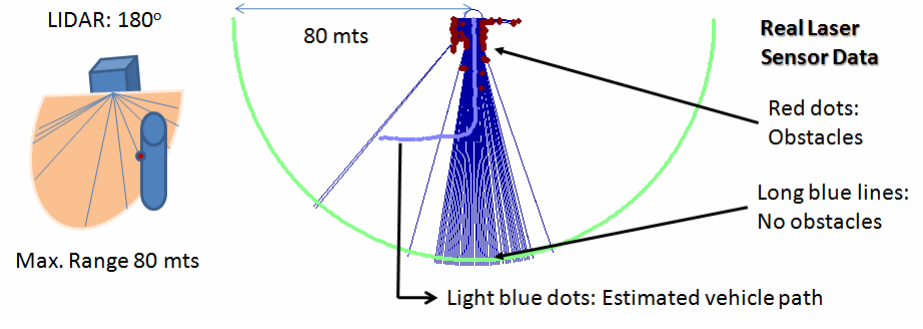


Fig. 3. LIDAR sensor data and obstacle detection

This data set allows us to estimate the vehicle path (as indicated in Fig. 3) and to focus the attention on potential obstacles present in this path. Note that several obstacles are detected by the laser sensor but many of them are not in the vehicle route. Due to this approach we are able to reduce significantly the number of false alarms of dangerous obstacles in the vehicle trajectory.

2.3 Obstacles Detection

The obstacle detection can occur in three ways: i) *Global Detection*: detect all the obstacles that are around the vehicle in the LIDAR sensorial range; ii) *Fixed Rectangular Region*: detect only the obstacles that are in the frontal area of the vehicle; iii) *Adaptive Region*: use an intelligent algorithm to detect only the obstacles that are in the frontal area of the vehicle and also inside the adaptive region obtained by data fusion and analysis of vehicle path.

Global Detection

As presented in Fig. 4(B), any obstacles that are in the laser sensorial range are detected and considered as potential harmful objects. Once the laser has an 180° wide scanning (see Figs. 3 and 4(A)), all the obstacles in a radius of 80 meters inside the ahead scanned area are detected. This approach is not practical since we detect several obstacles that do not represent a significant menace to the car neither to the driver, as for example, trees and lampposts that are side by side with the vehicle (not in the vehicle trajectory).

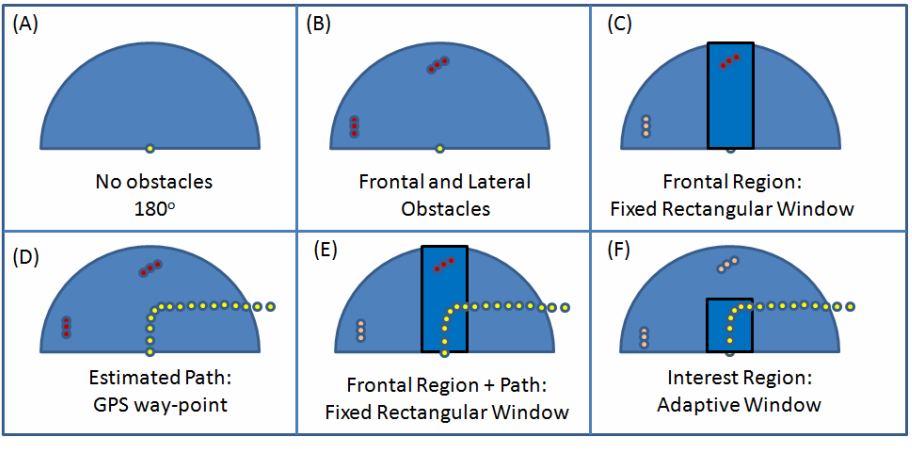


Fig. 4. Obstacles detection using sensor fusion, fixed and adaptive windows: (a) LIDAR visible region; (b) LIDAR detection of obstacles; (c) LIDAR only based fixed rectangular window for frontal obstacle detection; (d) LIDAR and GPS fusion: GPS estimated path (way-points); (e) False alarm caused by obstacles which are out of the estimated vehicle trajectory; (f): Sensor Fusion: adaptive window used to avoid false alarms.

Fixed Rectangular Region

As presented in Fig. 4(C), it is possible to define a rectangular bounding box in front of the vehicle. Only the obstacles that are inside of this rectangular window are considered as possible harmful obstacles. The user can set it up the dimensions of this fixed rectangular area in front of the vehicle, as for example, defining a rectangular

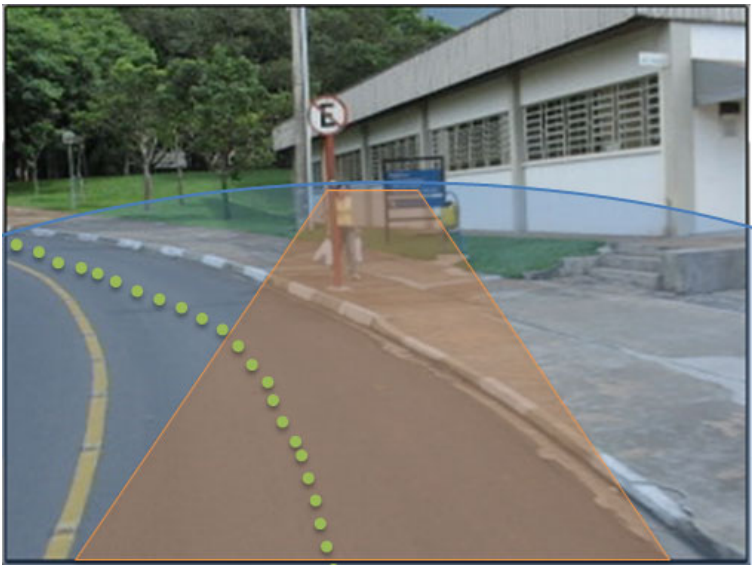


Fig. 5. Obstacles detection using a fixed rectangular region

area of 20m x 3m (Length x Width) which is used to limit the inspected sensorial area searching for possible obstacles. This approach reduces significantly the total number of detected obstacles.

The problem of adopting a fixed rectangular bounding box in front of the vehicle is the fact that usually the driver has a predefined path to be followed, and there are several other obstacles surrounding the vehicle, which are not blocking or harming the specified trajectory. For example, as presented in Fig. 4(D) and in a real situation during experiments (Fig. 5), when the vehicle turns, all the obstacles that are in front of the vehicle, after passing the turning point, should not be considered (e.g. trees, people crossing the street, buildings etc.). As it can be observed in Fig. 4(E), depending upon the vehicle trajectory, several obstacles can produce false alarms when present in front of the vehicle, even if these obstacles do not represent any danger related to the vehicle trajectory.

Adaptive Region

Our proposal is to intelligently integrate the GPS and Compass data that describes the vehicle estimated path within the LIDAR sensor data that are showing the obstacles present in the environment. The sensor fusion allows a better adjust of bounding box used to identify potential obstacles in the vehicle path. The Figure 4(F) presents an example of the rectangular bounding box resizing when the vehicle was preparing to turn right. Note that some obstacles are in front of the vehicle, but they are in a region out of the vehicle path. So, after the intelligent adjust of attention zone, they are not considered as possible harmful obstacles, resulting in a reduction in the number of false alarms.

2.4 Intelligent Attentional Algorithm: Adaptive Region Selection

The proposed algorithm is presented bellow (see Algorithm 1):

Algorithm 1. Adaptive Region Selection

Adaptive Region Algorithm	Inputs:	LIDAR Data: Vector [] – Angle/Distance measurement points; GPS Position: Latitude Longitude and Compass Heading
	Outputs:	GPS Way-Points: Vector [] - Lat., Long., Heading Obstacles: Vector [] – Angle/Distance measurement points

1. Find the closest GPS way-point (P*i*) related to the present GPS position (P*t*)
2. Considering '*N*' GPS way-points starting at Pi (from P*i* to Pi_{+N})
   - 2.1 Transform the absolute coordinates of P*i+j* (Lat., Long, Head.) to relative coordinates measuring distances and orientations relative to the frame of the LIDAR sensor.
   - 2.2 Plot the LIDAR range obtaining an half-circle (180^o x 80m radius)
   - 2.3 Plot each LIDAR Data according to Angle/Distance measured
   - 2.4 Plot the transformed GPS points P*i+j* inside the LIDAR range area
3. Considering the vehicle width W, create a rectangular bounding box, defining the maximum limits of each laser beam (max. distance)
   - 3.1 Find the last point *k* of the GPS points P*i+j* that remains inside the rectangle borders defined by the vehicle width W (left/right borders)
   - 3.2 Using the GPS point P*k,* obtain the maximum distance of this point related to the origin of the frame of the LIDAR sensor. Use this distance D*k* to adapt the length of the adaptive region selection
   - 3.3 Plot the rectangle defined by the vehicle width W and length D*k* inside the LIDAR range area
   - 3.4 Select all LIDAR data points that remain inside this rectangle and copy them to the Obstacles Vector (Output)

3 Experimental Results

The experiments have been carried out at the University campus. The path chosen for data collection corresponds to a very diverse circuit with 840 meters of campus internal streets (Fig 6). The passage comprised by points A and B is a straight street, with width equivalent to two cars and two-way orientation. In the left side there is an area reserved for upright vehicles parking, and in the right side, a walkway and a large building that occupies the entire block.

Between points B and C there is an open area with trees and some information signs. The segment between C and D is composed of crossing routes and also some tracks for pedestrian crossing. Section D and E is mostly a straight way with grass on both sides, while the last (E and F) is wide open. At the F point, there is an 180deg turn in front of the campus entrance.

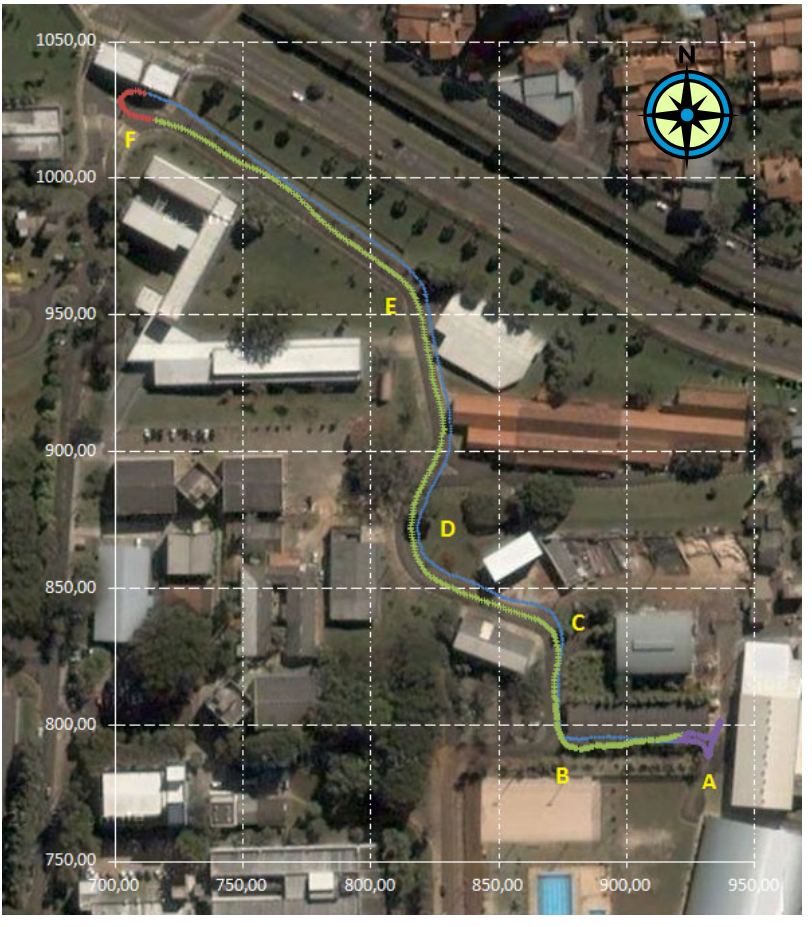


Fig. 6. Area used for the experimental tests

The Table 1 describes the experimental results obtained to route comprised by AEA segment:

Table 1. Experimental Results

DATASET 1	*Global Detection (GD)*	*Fixed Rect. Region (FRR)*	*Adaptive Region (AR)*
Total Data Points detected inside the attention area	5861	604	82
Total of Warnings for different obstacles	>> 100	27	2

Considering the tabled results, we can observe that the proposed method can drastically reduce the number of false alarms. Using the adaptive region we are able

to reduce from hundreds of possible harming obstacles to only 2 alarms of dangerous obstacles. When observing the *Global Detection* results (no fusion, all sensorial range), we get a large number of alarms and elements detected, as expected. The *Fixed Region Rectangle* Fig. 7 (left) where had no fusion, fixed rectangular bounding box of 3 meters width and 20 meters length, resulted in 604 collision data points, and estimated occurrence of 27 different alarms for unique obstacle count. The proposed approach of using *Adaptive Region* using a sensor fusion Fig. 7 (right), reduced the alarms to only one situation when 2 different obstacles were detected. Observing the video captured from the experiment, these two obstacles are really dangerous elements that are very close to the vehicle path, and the warning emitted in this situation probably will be well accepted by the driver.

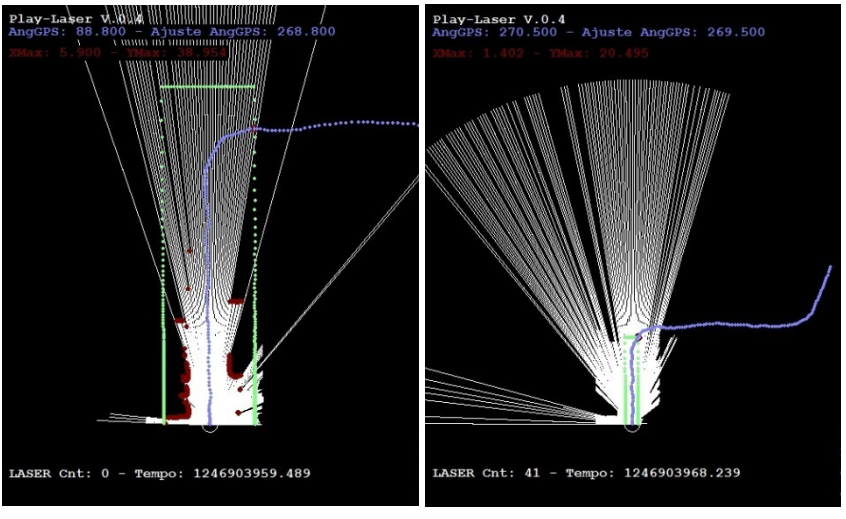


Fig. 7. Screenshots of DAS System taken during experiments

4 Conclusion

This work presented an approach to combine GPS and laser scanner data in order to focus the attention to significant possible collision data points that are in the vehicle estimated path. Through sensor fusion an intelligent algorithm can adjust the perception area and differentiate potential harmful objects of real dangerous obstacles. The proposed approach allowed a significantly reduction in the number of false alarms of possible collisions, as demonstrated in our practical experiments based on real data acquired using a vehicle equipped with LIDAR, GPS and Compass sensors.

Acknowledgments. The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology - Critical Embedded Systems - Brazil), processes 573963/2008-9 and 08/57870-9.

References

1. Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., et al.: Stanley: The robot that won the darpa grand challenge. Research Articles. J. Robot. Syst. 23(9), 661–692 (2006)
2. Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., et al.: Autonomous driving in urban environments: Boss and the urban challenge. J. Field Robot. 25(8), 425–466 (2008)
3. ELROB: The European Robot Trial, http://www.elrob.org/ (last visited June 2010)
4. Subramanian, R.: Motor Vehicle Traffic Crashes as a Leading Cause of Death in the U.S., 2002 – A Demographic Perspective. NHTSA Technical Report. Springfield, VA (June 2005)
5. Rossi, J. and Maia, H.: Mortes no trânsito se iguala a homicídios. Jornal da Tarde (06/18/2008), http://carros.wordpress.zap.com.br/ultimas-noticias/mortes-no-transito-se-iguala-a-homicidios-20080618/ (last Visited June 2010)
6. ITARDA: What Sort of Human Errors Cause Traffic Accidents? Institute for Traffic Accident Research and Data Analysis. Technical report. Japan (2001)
7. Keay, L., Munoz, B., Turano, K.A., et al.: Visual and Cognitive Deficits Predict Stopping or Restricting Driving: The Salisbury Eye Evaluation Driving Study (SEEDS). Investigative Ophthalmology and Visual Science 50, 107–113 (2009)
8. Kyo, S., Okazaki, S.: In-vehicle vision processors for driver assistance systems. In: Design Automation Conference, ASPDAC Seoul, Korea, pp. 383–388 (April 2008)
9. Agarwal, V., Murali, V., Chandramouli, C.: A Cost-Effective Ultrasonic Sensor-Based Driver-Assistance System for Congested Traffic Conditions. IEEE Transactions on Intelligent Transportation Systems 10(3), 486–498 (2009)
10. Sick Ag Inc.: Quick Manual for LMS communication setup, Germany (March 2002)
11. Garmin International Inc.: GPS 18x Technical Specification 190-00879-08 Revision B. Kansas, USA (January 2008)
12. True North Technologies LLC.: Revolution Compass User's Guide. Document No. 1554-D. Massachussets, USA (2008)

An Efficient Strategy for Fast Object Search Considering the Robot's Perceptual Limitations

Javier Cabanillas, Eduardo F. Morales, and Luis Enrique Sucar

Instituto Nacional de Astrofísica, Óptica y Electrónica,
Luis Enrique Erro 1, Tonantzintla, Puebla, Mexico
{javier.cabanillas}@ccc.inaoep.mx,
{esucar,emorales}@inaoep.mx
http://ccc.inaoep.mx

Abstract. Searching for an object in an environment using a mobile robot is a challenging task that requires an algorithm to define a set of points in which to sense the environment and an effective traversing strategy, to decide the order in which to visit such points. Previous work on sensing strategies normally assume unrealistic conditions like infinite visibility of the sensors. This paper introduces the concept of *recognition area* that considers the robot's perceptual limitations. Three new sensing algorithms using the recognition area are proposed and tested over 20 different maps of increasing difficulty and their advantages over traditional algorithms are demonstrated. For the traversing strategy, a new heuristic is defined that significantly reduces the branching factor of a modified *Branch & Bound* algorithm, producing paths which are not too far away from the optimal paths but with several orders of magnitude faster that a traditional *Branch & Bound* algorithm.

1 Introduction

One of the main goals in robotics is to design an autonomous robot able not only to move around an environment and to avoid objects, but also to interact with objects accomplishing a visibility-based task. In particular, this work is focused in the problem of searching for an static object in an known environment. Our goal is to generate a set of sensing locations and develop a motion strategy to visit this set to find the object in a reasonable time.

In any environment of n vertexes and h holes, $\lfloor (n+h)/3 \rfloor$ sensing locations are sufficient, and sometimes necessary, to see every interior point of the environment [4]. This set of sensing locations, however, is usually far from optimal in terms of cardinality. Finding the minimal number of sensing locations is an open problem called the Art Gallery Problem (AGP) [8]. This problem consists of minimizing the cardinality of the set of guards or sensing locations required to cover a polygon. There are different schemes to generate and to combine this set of sensing locations and there is a large number of algorithms that have been proposed related to the AGP and its variants to generate a set of sensing locations (e.g., [7,4]).

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 552–561, 2010.

There is also ample work on exploration and mapping within the area of rescue, search and surveillance robots, but generally it is focused on exploring an unknown environment, which is a different task (e.g., [6,2,1]).

In known environments, one approach to search for objects considers a robot continuously sensing the environment. In this approach, a simple Levy's flight or a greedy strategy using the gradient of the new visibility area can be used to guide the search [12]. An alternative approach considers only a set of specific locations where the robot senses the environment, which simplifies the task. In [9,10] the authors assume that a set of sensing locations is given as input. In order to solve the search problem, they used a heuristic based on an utility function, defined by the probability of see the object in the location over the distance traveled. A methodology that takes into account the problem of recognition is proposed in [11]. In order to achieve a reliable recognition, an attention mechanism based on receptive field co-occurrence histograms and SIFT feature matching is proposed. The authors also present a method for estimating the distance to objects. In this work, however, the set of sensing locations is also given as input.

A common weakness of current approaches that work with sensing locations is that they assume infinite visibility for the sensors, as define in the classic AGP [7]. This is clearly an unrealistic assumption for the sensors used in real robots. In this paper, we introduce the concept of *recognition area* that depends on the sensing capabilities of the robot and the characteristics of the object.

Different sensing locations strategies have been proposed in the past. In this paper, we incorporate the recognition area concept into three new sensing location algorithms. The first algorithm improves over a triangulation-based algorithm that generates a smaller number of points and also moves further away from obstacles. The second algorithm incorporates a pre-processing method to a probabilistic road map (PRM) algorithm to significantly improve its performance. The third algorithm is based on a random generation process followed by a greedy search strategy to very quickly generate sensing paths.

Once the set of sensing locations has been defined, they have to be visited in a specific sequence in order to minimize the total traveled distance to find the object. This combinatorial problem is solved using a new heuristic that reduces the branching factor of a *Branch & Bound* algorithm. The results show that the distance traveled is not far from the optimal and the run time is up to 1000 times faster in the tested maps than a standard *Branch & Bound* algorithm.

The rest of this paper is organized as follows: Section 2 presents the terminology and main concepts used in this paper, Section 3 introduces the proposed solutions both for generating sensing points and for creating a traversing strategy. The experiments and results are described in Section 4. Finally, conclusions and future research directions are given in Section 5.

2 Problem Definition

Polygons are a convenient representation for many real-world environments because a polygon is often an accurate-enough model for many real objects and

polygons are easily manipulated in computational terms [8]. Therefore, it is assumed that the environment A is known and modeled as $P-H$ (see Figure 1 (b)), where P is an external polygon of the environment and $H = h_1, h_2, .., h_i, .., h_n$ is the set of internal polygons or holes (see Figure 1 (a)). If $|H| = 0$, P is considered as *simple*. If P has one or more holes, any hole in P is *simple*, which means that no hole can have holes within itself. The robot is considered as a simple polygon R and Minkowsky sums of (R, h_i), $\forall h_i \in H$ are used to get a workspace W that takes into account the geometric restrictions of R and the shape of A [5] (See Figure 1 (c)).

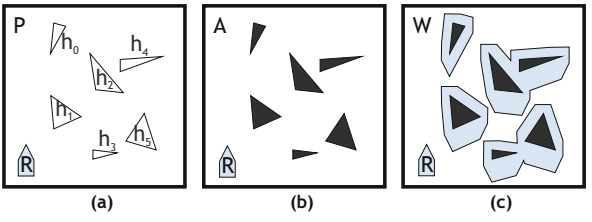


Fig. 1. (a) The exterior polygon P and the set of interior polygons $H = h_0, h_1, h_2, h_3, h_4, h_5$. (b) The environment $A = P - H$. (c) The calculated workspace $W = MinkowskySum(R, h_i) \forall h_i \in H$.

Since $Area(W) \subset Area(A)$, in the infinite visibility model, the largest line segment inside the visibility polygon of any point $p \in W$ is not larger than the largest axis of the bounding box of A. This allows to define a limited radius for an infinite range camera sensor. However, this is not enough to overcome the limitations of the model as shown in Figure 2.

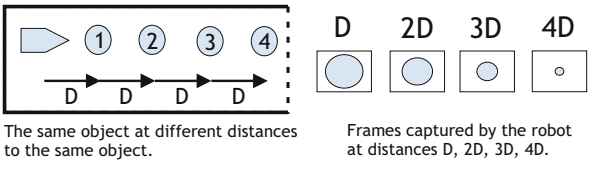


Fig. 2. If the distance to the object continues to increase, the object can be impossible to identify by the robot

Considering infinite visibility, the object finding problem ends when the object is within the visibility polygon of the observation point p. However, in practice, even if the object is inside the visibility radius from p, the recognition algorithm can fail when the distance to the object is too large or too small. This is like having at the same time a myopic robot, that cannot recognize objects that are too far away, and a robot with astigmatism that cannot recognize objects that are at very short distances.

Definition 1: The *recognition area* of a sensor s is defined by the minimum (d_{min}) and maximum (d_{max}) distances at which s is able to reliably recognize an object.

The size of the recognition area depends on the quality of the sensors and is an adjustable parameter of the proposed algorithms. We simulated different sizes of recognition areas in the reported experiments, however, it can be estimated by moving a robot from a long distance towards an object. Although the recognition algorithm can fail even at the recognition area, in this paper it is assumed that the sensor is "perfect" within this range.

After the generation of the set of sensing locations, G, it is still necessary to implement a strategy to visit all of them. The robot starts at a particular location, $g_0 \in G$, and visits the other locations as time progresses. The objective in this step is to find the route that minimizes the traveled distance to find the object. The searching task finishes when the object is inside the recognition area of the actual robot location.

3 Proposed Solution

The proposed solution has three steps: (i) compute the workspace using Minkowsky Sums, (ii) find a set of sensing locations that covers as much as possible the environment in a reasonable time considering the recognition area of the sensors, and (iii) find a traversing strategy in a reasonable time that minimizes the total travelled distance. The complete strategy is shown in Figure 3.

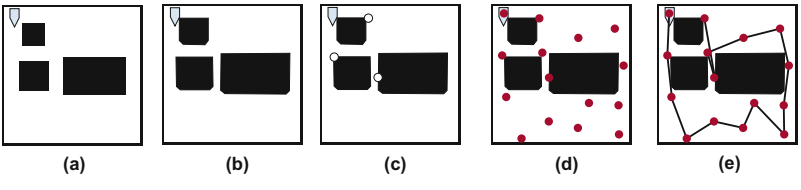


Fig. 3. The figure shows the complete strategy and its application in a simulated environment. The path is computed in 0.14 seconds for the square environment of 512 units of side and 100 units of visibility radius. (a) Robot and environment. (b) Minkowsky Sums. (c) Generation of a candidate set. (d) Generation of the final set. (e) Computation of the path.

3.1 Sensing Locations

In this paper we introduce three algorithms for sensing location based on: a) polygon partition, b) Probabilistic RoadMaps and c) a completely random process and a greedy strategy. Their advantages and disadvantages of each one are analyzed with 20 different maps.

Algorithm based on Triangulation. The decomposition of a simple polygon into triangles is called a triangulation. Various algorithms have been developed for triangulation, each characterized by its asymptotic order as n grows without bound. The simplest algorithm, called Ear Clipping, is the algorithm used in this paper. Algorithms with better asymptotic order exist, but are more difficult to implement [3]. Ear Clipping was designed for simple polygons but can also be applied to polygons with holes by following the process described in [3]. To improve its performance, a candidate set C is built as the union of the set of vertexes (external and internal) and the set of circumcenters (see Figure 4).

The goal is to generate a visibility or guarding set $G \subset C$ that is as small as possible; it is expected for $|G|$ to be much smaller than $|C|$. The algorithm greedily adds candidates using a scoring function to the guarding set G, until the entire polygon is covered. The score, $\sigma(c)$, for a candidate $c \in C$ is the number of points in C that are seen from c and that are not already seen by a point in G. At each iteration of the algorithm, the candidate c with the highest score $\sigma(c)$ is added to G, and the scores $\sigma(*)$ are updated accordingly. The algorithm stops when all the $c \in C$ are considered or when A is totally covered. The proposed algorithm has two main advantages over the basic triangulation algorithm: (i) It generates a smaller number of sensing points and (ii) the generated points tend to be farther away from obstacles. However it needs more computational time than the classic triangulation algorithm.

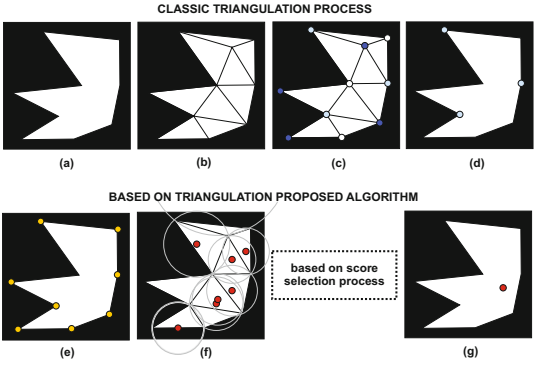


Fig. 4. The figure shows the proposed triangulation process. (a) The polygon. (b) Triangulate the polygon. (c) 3-Colored Algorithm. (d) Selected sensing locations of the basic algorithm. (e) The candidate set 1. (d) The candidate set 2. (f) The selected sensing location after the greedy algorithm process.

Improved Algorithm Visib-PRM. Roadmaps are usually used to solve motion planning problems, but in this case, it is used to obtain a set of sensing locations in the collision-free space (i.e., $P - H$). Due to the probabilistic completeness of the PRM method, we are sure that eventually G will complete the connectivity of the free space, but to prevent exorbitant running times, normally the node generation process is guided by heuristics that create higher

node densities in difficult areas. In this paper a list of points G_i of least visibility that are closer to obstacle edges and around the convex obstacle corners is obtained and one point by each obstacle is selected by its respective area. This preprocessing step incorporated into the original algorithm significantly boosts the performance of the generation set algorithm. Figure 5 show the proposed Visib-PRM algorithm.

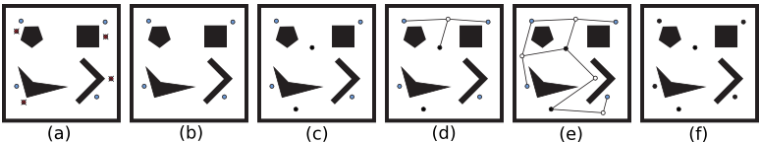


Fig. 5. (a) The selection process of the preprocessing step. (b) G_i (c,d,e) Application of Visib-PRM algorithm. (f) The final set of guards.

It is easy to adapt the original algorithm to work with a recognition area because although the basic algorithm assumes infinite visibility conditions, when it is applied in a domestic environment, this area of visibility is limited by the walls. Therefore, we can easily simulate restricted visibility conditions using a tunable parameter that simulate walls and that depends on the quality of the sensors.

Roma Algorithm. The proposed version of Visib-PRM is a partially random algorithm. Its principal weaknesses is that it depends on the environment and the selection process can need significant computational resources. The Roma algorithm is a very fast random-based algorithm combined with a greedy strategy that works as follows:

1. Generate a random point on the map.
2. Compute the covered and uncovered regions. This is shown in Figure 6 and can work both with an infinite or limited recognition areas.
3. Select the region of largest area and obtain its bounding box.
4. Generate a new random point until it is inside the previously computed bounding box.
5. Repeat the process until the number of generation attempts is equal to a threshold value M or the map is completely covered.

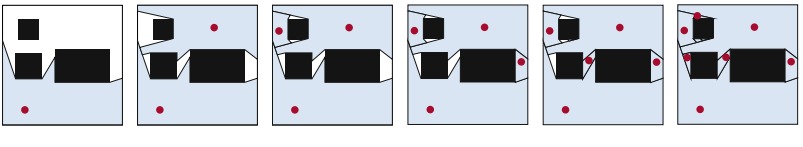


Fig. 6. The figure shows the generation process of ROMA algorithm using infinite visibility

Randomized points On the MAp (ROMA): (i) can work an order of magnitude faster than the previous algorithms, (ii) does not use a candidate set, therefore it avoids the selection process of the previous algorithms, (iii) it generates points only in the configuration free-space, and (iv) its dependency on the map complexity is lower than the partition based algorithms.

3.2 Motion Planning

Given a set of sensing locations we need a strategy to visit them. We know the distances between the sensing locations, and the robot has to visit each location exactly once until the last location is reached or the object is found. This is equivalent to a Hamiltonian Path problem that can be solved using a *Branch & Bound* algorithm. In its simplest form *Branch & Bound* can follow a LIFO strategy, a FIFO strategy or a Lowest Cost strategy.

In this paper we present a modified *Branch & Bound* algorithm as the required time to compute a path grows exponentially with the number of nodes[1]. Our solution includes the definition of a new heuristic that significantly reduces the branching factor and consequently the computing time.

Definition 2: Let L be the set of sensing location points. A location L_j strictly dominates another location L_k if and only if the two following conditions are true: (i) The area of the recognition polygon of L_j is greater than the area of the recognition polygon of L_k and (ii) the distance of $L_i \in L$ to L_j is less than the distance of L_i to L_k.

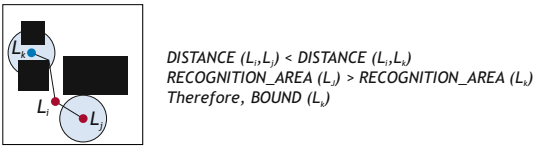


Fig. 7. The sensing location L_k is bounded because it is strictly dominated by L_j

The algorithm only considers non dominated nodes (see Figure 7):

- For the last location along the current solution (initially just the robot's starting location) explore the possible routes (create a new level in breadth-first search)
- For each node that needs to be expanded, compute the set of locations that are not strictly dominated by others and only choose those as children. This can be done with a convex hull algorithm in $O(nlog(n))$.
- Choose the best leaf according to the cost function, and start over with the best node as root.

[1] We were not able to obtain paths in a reasonable time with more than 13 nodes in a standard laptop.

4 Simulation Results

The proposed approach was tested in 20 different maps of increasing complexity (with and without holes), some of which are shown in Figure 8.

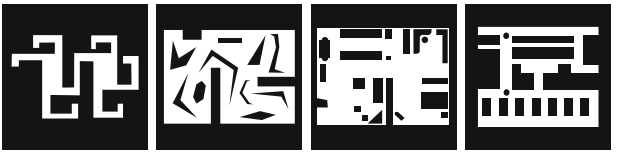


Fig. 8. Some of the maps used for testing

We measure the performance of the different sensing location algorithms in terms of: (i) number of vertexes of the map, (ii) number of guards generated, and (iii) computing time. We compared the proposed algorithms with their traditional counterpart algorithms and between each other (see Figure 9 and 10).

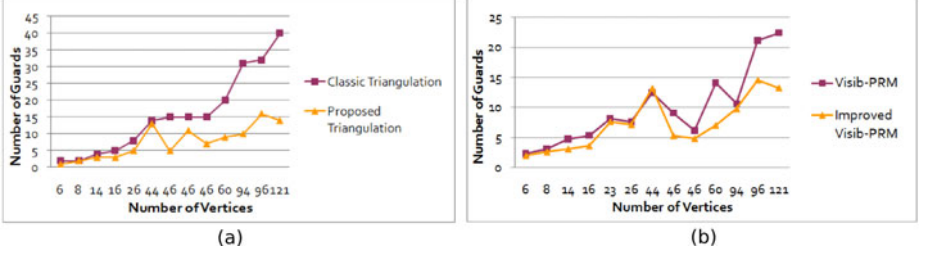


Fig. 9. For the different maps (the number of vertexes in each map is shown): (a) Comparison between the proposed triangulation algorithm and its traditional counterpart and (b) the same for the Visib-PRM algorithm

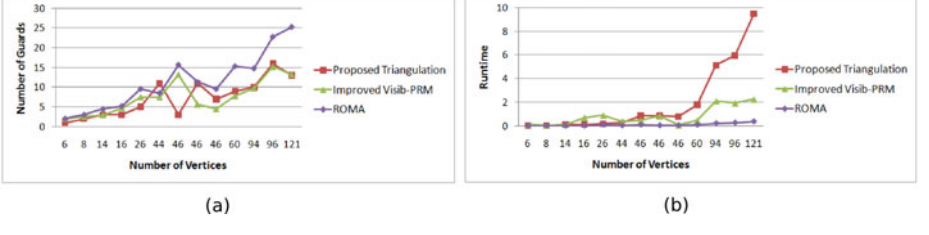


Fig. 10. For the different maps: (a) Comparison of the run-time under the number of vertexes. (b) Comparative of the number of guards under the number of Vertexes.

As can be seen from the figures, the proposed algorithms generate a smaller number of sensing points. On the other hand, when comparing the three new

algorithms, it can be seen that the ROMA algorithm is the fastest one followed by the Visib-PRM algorithm. Also, the Visib-PRM algorithm offers a good compromise between the number of sensing locations and computational time.

We also performed experiments of our algorithms with infinite visibility and limited visibility for different restricted area sizes using the Visib-PRM algorithm. As expected the number of sensing points increases with more restricted visibility conditions (see Figure 11), however, it is very competitive for reasonable ranges.

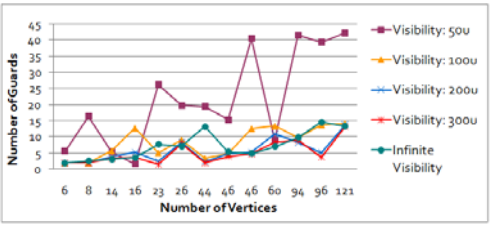


Fig. 11. Comparison between different ranges of visibility using Visib-PRM

We also compared the performance of our *Branch & Bound* algorithm against a standard *Branch & Bound* algorithm in terms of total distance of the generated paths and computing time. The results are shown in Figure 12 (note the log-scale in the run-time figure on the left).

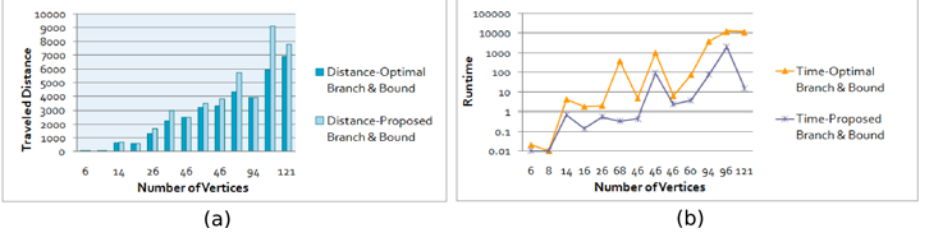


Fig. 12. Comparison between the proposed B&B algorithm and the optimal solution in terms of (a) distance and (b) time

As can be appreciated, the proposed algorithm is reasonably competitive in the total travelled distance when compared with the optimal solution (with 23% more distance in the worst case), but it is significantly faster (more than 3 orders of magnitude in the best case).

5 Conclusions and Future Work

In this paper we have introduced several strategies to solve the problem of searching an object in indoor environments. The problem involves two steps: (i) finding

a set of sensing points that cover all the area and (ii) designing a strategy to visit them. We introduce the concept of recognition area to deal with more realistic conditions, and incorporate it into three new sensing point generation algorithms. For the path planning step we introduce a new heuristic to significantly reduce the branching factor and reduce the processing time. The experimental results show the benefits of the proposed algorithms in terms of processing time, realistic conditions and number of sensing points. We are currently working in the incorporation of the above strategies into a real robot. We are also planning to consider information about the prior probability of object presence inside the recognition area to modify and improve the searching process, starting by the most probable locations. This information could be used as domain knowledge or as global object features, such as color or shape, to focused the sensing locations on promising areas.

References

1. Baxter, J.L., Norman, M.: Multi-robot search and rescue: A potential field based approach
2. Casper, J., Murphy, R.: Human-robot interaction during the robot-assited urban search and rescue effort at the world trade center. IEEE Transactions on Systems, Man and Cybernetics Parts B 33(3), 367–385 (2003)
3. Eberly, D.: Triangulation by ear clipping (March 2008)
4. Hoffmann, F., Kaufmann, M., Kriegel, K.: The art gallery theorem for polygons with holes. In: The 32nd IEEE Symposium on the Foundation of Computer Science, pp. 39–48 (1991)
5. Latombe, J.C.: Robot motion planning, Boston (1991)
6. Nourbakhsh, I.R., Sycara, K., Koes, M., Yong, M., Lewis, M., Burion, S.: Human-robot teaming for search and rescue. IEEE CS and IEEE ComSoc (2005)
7. O'Rourke, J.: Art gallery theorems and algorithms. Oxford University Press Inc., New York (1987)
8. O'Rourke, J.: Computational Geometry in C. Cambridge University Press, Cambridge (1998) Hardback ISBN: 0521640105; Paperback: ISBN 0521649765
9. Sarmiento, A., Murrieta-Cid, R., Hutchinson, S.: An efficient strategy for rapidly finding an object in a polygonal world. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1153–1158 (2003)
10. 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 International Conference on Robotics and Automation, pp. 3497–3502 (2005)
11. Sjöö, K., Galvez-Lopez, D., Paul, C., Jensfelt, P., Kragic, D.: Object search and localization for an indoor mobile robot. Journal of Computing and Information Technology - CIT 17(1), 67–80 (2009)
12. Tovar, B., LaValle, S.M., Murrieta-Cid, R.: Optimal navigation and object finding without geometric maps or localization. In: Proc. IEEE International Conference on Robotics and Automation, pp. 464–470 (2003)

Building Feature-Based Maps with B-splines for Integrated Exploration

Alfredo Toriz P.[1], Abraham Sánchez L.[2], Rene Zapata[1], and Maria A. Osorio[2]

[1] LIRMM, UMR 5506
161, rue Ada, Montpellier - France
[2] Computer Science Department
Benemérita Universidad Autónoma de Puebla
alfredo.torizpalacios@lirmm.fr, asanchez@cs.buap.mx, zapata@lirmm.fr

Abstract. This paper describes a simultaneous planning localization and mapping methodology, where the robot explores the environment efficiently and also considers the requisites of the SLAM (Simultaneous Localization And Mapping) algorithm. The method is based on the randomized incremental generation of a data structure called Sensor-based Random Tree, which represents a roadmap of the explored area with an associated safe region. A continuous localization procedure based on B-Splines features of the safe region is integrated in the scheme. The approach is evaluated for accuracy and consistency using computer simulations and for effectiveness using experimental data from different real environments.

1 Introduction

SLAM approaches are used simultaneously with classic exploration algorithms. However, results obtained with SLAM algorithm strongly depends on the trajectories performed by the robots. Classic exploration algorithms do not take localization uncertainty into account, when the robots travel through unknown environments, the uncertainty over their position increases and the map construction becomes more difficult. Consequently, the result obtained can be a useless and inaccurate map. With the integrated exploration or SPLAM (simultaneous planning localization and mapping) algorithm proposed, robots can explore the environment efficiently and the requisites of SLAM algorithm can be also considered.

An integrated exploration method is introduced in [1] to achieve the balance of speed of exploration and accuracy of the map using a single robot. To minimize the overall exploration time of robots in an unknown environment, each robot should be moving in a direction and at a speed that maximize the exploration areas. Freda et al. [2] use a sensor-based random tree (SRT). The tree is expanded as new candidate destinations near the frontiers of the sensor coverage are selected. These candidate destinations are evaluated considering the reliability of the expected observable features from that points. The tree is used to

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 562–571, 2010.

navigate back to past nodes with frontiers when no frontiers are present in the current sensor coverage.

Recently, a novel laser data based SLAM algorithm using B-Spline as features has been developed in [3]. EKF is used in the proposed BS-SLAM algorithm and the state vector contains the current robot pose together with the control points of the splines. The observation model used for the EKF update is the intersections of the laser beams with the splines contained in the map.

The basics of the B-splines are briefly presented in Section II. The proposed approach to solve the simultaneous planning localization and mapping problem is detailed in Section III. Simulation results are discussed in Section IV. Finally, conclusion and future work are detailed in Section V.

2 Fundamental of B-splines

Let U be a set of $m+1$ non-decreasing numbers, $u_0 \leq u_2 \leq u_3 \leq \ldots \leq u_m$. The u_i's are called knots, the set U the knot vector, and the half-open interval $[u_i, u_{i+1})$ the i-th knot span. Note that since some u_i's may be equal, some knot spans may not exist. If a knot u_i appears k times (i.e., $u_i = u_{i+1} = \ldots = u_{i+k-1}$, where $k > 1$,u_i is a multiple knot of multiplicity k, written as $u_i(k)$. Otherwise, if u_i appears only once, it is a simple knot. If the knots are equally spaced (i.e., $u_{i+1} - u_i$ is a constant for $0 \leq i \leq m-1$), the knot vector or the knot sequence is said uniform; otherwise, it is non-uniform. The knots can be considered as division points that subdivide the interval $[u_0, u_m]$ into knot spans. All B-spline basis functions are supposed to have their domain on $[u_0, u_m]$. To define B-spline basis functions, we need one more parameter, the degree of these basis functions, p. The i-th B-spline basis function of degree p, written as $N_{i,p}(u)$, is defined recursively as follows:

$$N_{i,0}(u) = \begin{cases} 1, \text{ if } u_i \leq u \leq u_{i+1}; \\ 0, \text{ otherwise.} \end{cases}$$
$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u) \tag{1}$$

The above equation is usually referred to as the Cox-de Boor recursion formula [4]. Given $n+1$ control points $P_0, P_1, ..., P_n$ and a knot vector $U = u_0, u_1, ..., u_m$, the B-spline curve of degree p defined by these control points and knot vector U is

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \tag{2}$$

where $N_{i,p}(u)$'s are B-spline functions of degree p. The form of a B-spline curve is very similar to that of a Bézier curve [5].

3 Our Approach

The strategy adopted to make the exploration process is named SRT, which is based on the construction of a data structure called Sensor-based Random

Tree (SRT) that represents the roadmap of the explored area with a safe region associated (SR); each node of the tree ($\mathcal{T}$) consists of a position of the robot and its associated local safe region (LSR) constructed by the perception system of the robot. A continuous localization process is performed using a newly developed method based on the comparison of environmental characteristics such as turns and curves compared with the new environmental curves extracted from the LSR current position. The process will be described later. The algorithm implemented for integrated exploration is shown down. In each iteration, the algorithm first gets the LSR associated with the current configuration of the robot, q_{curr}. Once obtained, the function EXTEND_TREE will be responsible for updating the tree by adding the new node and its LSR. Besides, the vector representing the curves that constitutes the environment will be updated by adding new curves drawn from the LSR that are not still part of it. The next step is to process the local frontier $\mathcal{F}$, where obstacles and free areas are identified.

Generally, $\mathcal{F}$ is a collection of discrete arcs. After obtaining these boundaries and if there are still free zones, the procedure RANDOM_DIR generates random directions in order to get one within the free arc. Then a q_{cand} configuration is generated, giving a step in the direction θ_{rand}. The step size is chosen as a fixed fraction of the radius of the LSR in that particular direction. Consequently q_{cand} will be collision-free due to the shape of S. If no free boundary arc is found, the robot will go back to the position of the parent node, q_{curr} and the exploration cycle will begin again.

Once the configuration q_{cand} is obtained, the procedure VALID_CONF will make sure that this new configuration is valid, i.e. that this new position is outside the LSR's of other nodes in the tree. If this new configuration is valid, it will be the new target configuration q_{dest} for the robot, however, if after a maximum number of attempts it is not possible to find a configuration q_{cand}, the parent node q_{cand} will be the new q_{dest} configuration and the robot will go back to the its parent configuration. Once the configuration q_{dest} is obtained, the function MOVE_TO(q_{curr}, q_{dest}) will move the robot to this configuration as follows : First, it will look in a previously defined list of control inputs ($list\_U$), an entry that approximates the robot's position to the position q_{dest} from the position q_{curr}, named $u_{control}$. Once obtained, $u_{control}$ will be applied to the robot, resulting in the displacement of the robot near to the goal. At this point, the odometric position stored in the memory of the robot and the increase in x, x and θ between the old and the current odometric position (x, y, θ) will be obtained again. This information, reported by the robot will be essential to get the position detected by the method Feature-Based Localization with B-splines proposed in this paper. The algorithm will be repeated until q_{curr} and q_{dest} configurations are the same.

This localization procedure uses B-spline curves to represent the frontiers between the free regions and the obstacles of a complex environment. It obtains in the first instance the estimated position of the robot, adding to the last position the position increments $\triangle x, \triangle y$ and $\triangle\theta$ made by the robot from the previous odometric position to the current odometric one; then the estimation of

```
INTEGRATED_EXPLORATION(q_init, K_max)
1  q_curr ← q_init;
2  for k=1 to K_max
3   S ← LSR(q_curr);
4   T ← EXTEND_TREE(q_curr, S, T);
5   ENV_BS←UPDATE_BSPLINES(S);
6   F ← FRONTIER(q_curr, S);
7   if F ≠ ∅
8    i ← 0
9    VALID ← FALSE
10   While((i < MAX_ITER) && (!VALID))
11      θ_rand ← RANDOM_DIR(F);
12      q_cand ← DISPLACE(q_curr, θ_rand);
13       VALID ← VALID_CONF(q_cand);
14       i++;
15   end
16   if (VALID)
17       q_dest ← q_cand;
18   else
19       q_dest ← q_curr.parent;
20       if q_dest =NULL
21          return [T,ENV_BS];
22       end
23  end
24  else
25   q_dest ← q_curr.parent;
26   if q_dest =NULL
27       return [T,ENV_BS];
28   end
29  end
30  MOVE_TO(q_dest, q_curr,ENV_BS);
31   q_curr ← q_dest;
32 end
33 return [T,ENV_BS];
```

Fig. 1. SRT-based integrated exploration algorithm

the surrounding environment captured by the sensor is obtained and the robot is located in the estimated position in the space. The sensor data are subject to two segmentations:

- FIRST SEGMENTATION. An analysis of the relative position of consecutive data points is performed. The aim is to detect points close enough that belong to the same obstacle.
- SECOND SEGMENTATION. The obtained segments in the first segment are again subjected to testing for consecutive points whose angle is below a certain threshold. The objective of this segmentation is to detect corners.
- SETTING. Each of the obstacles of the second segment will be adjusted to the B-spline of degree 3 that constitutes its control polygons.

Once the data from the sensor are segmented, a data association process is performed. A rough association is performed first; the control points of each segment are compared with the control points contained in the map using the following criteria: $\min(dist(X_{m,i}, X_{o,j})) \leq d_{min}$, $i = 1, ..., n_m$ and $j = 1, ..., n_o$. Where $X_{m,i}$ are the control points of the spline on the map and $X_{o,j}$ are the control points of the observed spline; n_m and n_o are the number of control points of the spline on the map and the detected spline respectively. The $dist(X_{m,i}, X_{o,j})$ represents the Euclidean distance between the control points and d_{min} is a parameter that regulates whether or not the points are associated. If no spline in the map is sufficiently close to a detected spline that can be related, the new object is added to the map after the position of the robot is located. By contrast, if a spline is associated with a feature of the map, then it is necessary to obtain the concordance between their points, this process executes as follows:

- One end of the curve is considered (**point a**).
- The closest point between the spline on the map and the **point a** is calculated (**point b**).
- If **b** is one of the spline's extreme points at the end of the map, then the closest point to **b** in the spline is calculated (**point c**) if not, the point is associated with **point b**.
- The process is repeated starting in the other end of the spline (**point d** in the figure 2), that is associated with the **point e** on the spline in the map.
- Due to the a B-splines property, the length of the curves can be known, and segments **e-b** and **d-c** can be adjusted to have the same length. If the difference of the lengths is greater than a threshold l_{max}, the extreme elements of the larger curve are eliminated to adjust its size.

Once the curves of the estimated position are associated with the environment curves, a final verification of the association must be performed. The distance of each of the ends of each curve to the other is obtained as shown in the figure below:

If the distances between the ends of each curve with the curve obtained with the B-splines of the environment are less or equal to certain threshold, there is a good partnership that will predict a useful localization (da, 1 = dn, 1 da, 2 = dn,

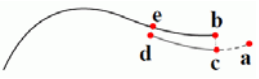


Fig. 2. Concordance between curves

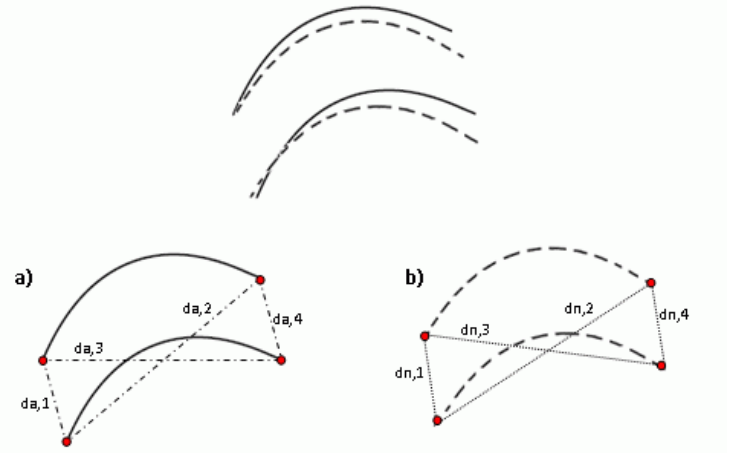


Fig. 3. Verification of curves association: a) Distance from the ends of the curves belonging to the environment. b) Distance from the ends of the curves belonging to the estimated position.

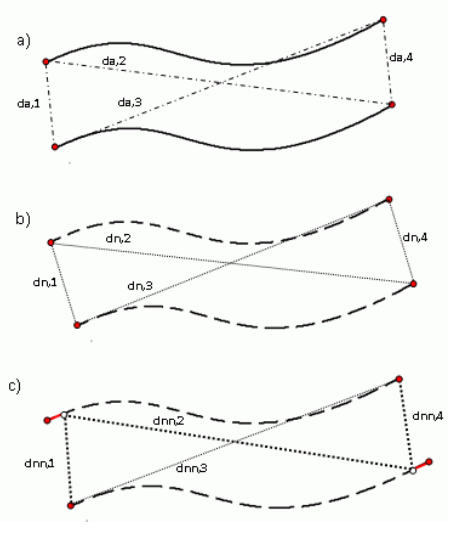


Fig. 4. Process of adjusting curves. a) Value of distances between the curves of the environment. b) A list of distances between the curves of the estimated position. Related distances are obtained: da,1 < dn,a, da,2<dn,2, da,3=dn,3 da,4<dn,4. c) Readjustment process where white nodes are the new ends and the red sections are eliminated, obtaining da,1=dnn,1, da,2=dnn,2, da,3=dnn,3,da,4=dnn,4.

2nd, 3 = dn, 3 da, 4 = dn, 4), otherwise the distances must be adjusted, eliminating the extreme elements of the curves whit the larger distances.

The association curves and their endpoints are used to correct the estimated configuration error in two stages (first angle and then translation) as follows. Let $q_{curr} = (x_{curr}, y_{curr}, \theta_{curr})$ be the current robot configuration and $q = (x, y, \theta)$ the obtained estimation with increments $\triangle x, \triangle y, \triangle \theta$.

$$\begin{aligned} x_{curr} &= \hat{x} + ex \\ y_{curr} &= \hat{y} + ey \\ \theta_{curr} &= \hat{\theta} + e\theta \end{aligned} \tag{3}$$

where ex, ey and $e\theta$ are the estimation errors. The angular correction $e\theta$ is processed according to the following procedure. Using the partnership data, all the features belonging to the stored map are extracted, all possible pairs of these characteristics are joined by line segments (completing a graph of the property), and the vector of angular coefficients of these segments is processed. The procedure is repeated for the characteristics relating to the estimation of the current position vector that will clearly be different due to the presence of an estimation error. The idea is then to correct the estimated orientation $\hat{\theta}$ in such a way that the norm of the difference between the two vectors are minimized in a least-squares sense.

Assume now that the above process is performed with the data stored in the association system and the result corresponds to the stored map; this result is denoted as α_{ij}^{ref}. To process the corresponding value of α_{ij}^{curr}, it is necessary to use the estimated coordinates of the current characteristics that correspond to the position S and $\theta_{curr} = \hat{\theta} + e\theta$ instead of θ. Therefore, α_{ij}^{curr} will depend only on $e\theta$, and we can identify the optimal correction e_θ^* can be identified as:

$$e_\theta^* = arg\ min_{e\theta} \sum_{i=1}^{N_a} \sum_{j=i+1}^{N_a} c_{ij} (\alpha_{ij}^{ref} - \alpha_{ij}^{curr})^2 \tag{4}$$

The weight c_{ij} depends on the segment's length (the direction of shorter segments is more sensitive to displacements of the endpoints) and the reliability of the pairs of the property. After the correction angle e_θ^* is obtained, the translation adjustments, ex and ey, are processed using again the complete feature graph generated above. Particularly, denoted by $M_{ij}^{ref} = (x_{ij}^{ref}, y_{ij}^{ref})$ the midpoint of the generic segment of the property in S_{ref}, and $M_{ij}^{curr} = (x_{ij}^{curr}, y_{ij}^{curr})$ is the corresponding midpoint in S. Clearly, for a given $e\theta$ the coordinate $(x_{ij}^{curr}, y_{ij}^{curr})$ depend only on ex and ey, respectively. Therefore, the best estimates of these translation adjustments are obtained as:

$$e_x^* = arg\ min_{ex} \sum_{i=1}^{N_a} \sum_{j=i+1}^{N_a} c_{ij} (x_{ij}^{ref} - x_{ij}^{curr})^2 \tag{5}$$

$$e_y^* = arg\ min_{ey} \sum_{i=1}^{N_a} \sum_{j=i+1}^{N_a} c_{ij} (y_{ij}^{ref} - y_{ij}^{curr})^2 \tag{6}$$

In principle, the angular correction can be performed using a single segment (i.e., a single curve segment), and the use of the characteristics complete graph is useful for the incorporation of redundant information that can strengthen the correction process of the erroneous data.

4 Experimental Results

A simulated robot and the real Pioneer P3DX robot equipped with front and rear bumper arrays, a ring of eight forward ultrasonic transducer sensors (range-finding sonar) and a Hokuyo URG-04Lx laser range finder were used in the experiment. The Pioneer P3DX robot is an unicycle robot. The LIRMM environment was used in the experimental and simulation tests (the environment had several corridors). Figure 5 show the two final maps: at the left, the obtained map without localization (only with odometric estimates); at the right, the obtained map with the proposed approach. Comparing the two final maps, it can be said that the robot did not use the localization process, collided frequently with the obstacles as can be appreciated in the left image.

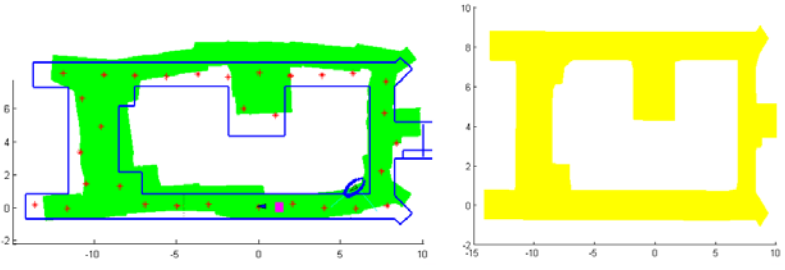


Fig. 5. Final map only with odometric estimates and the final map with localization

For the association process, the basic procedure proposed in [3] was retaken, but with a new functionality derived from the properties of the B-splines, i.e., we ensure that these new curves and those belonging to the environment have the same lengths. This new feature enabled better and more accurate association of the data collected by the sensor than the association obtained with the basic method originally proposed. The geometric properties of the ends of the curves was also considered, i.e., one can make a final check of the association taking the distances between the ends of the curves of the environment and the ends of the curves corresponding to the estimated position and verify their similitude. This exhaustive verification is necessary because the nature of the proposed localization method. It can be said that the approach presented made a good use of the parametric representation of the environment characteristics at the time of the data association.

Due to the exploration system used, in which each node in the exploration tree contained a robot position with its associated LSR, and the features of the

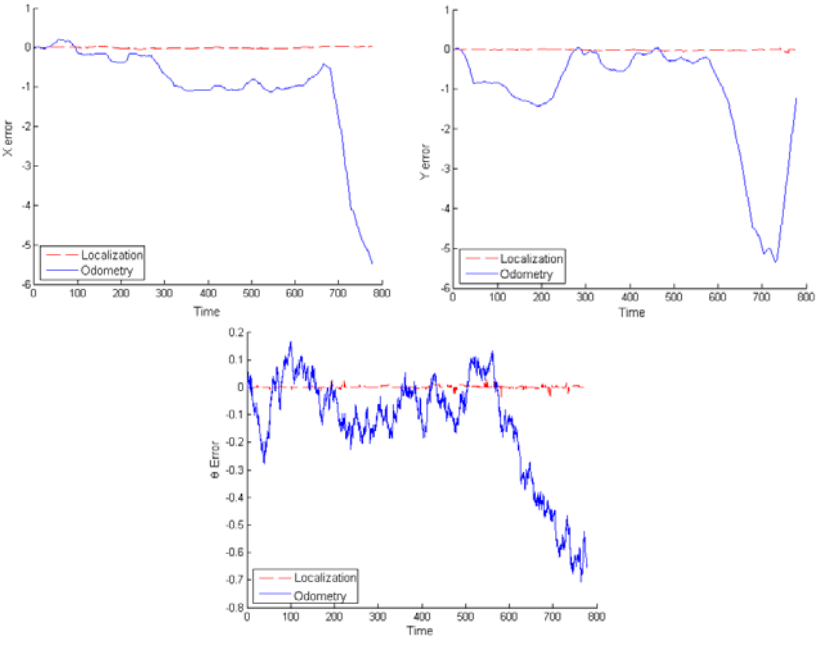


Fig. 6. x, y, θ errors for the odometry and the proposed approach

localization method, the map updating is performed every time a new LSR is obtained, unlike the approach presented in [3]. The updating of the environment is not performed in every robot's movement, but when a new position in the frontier between the known and the unknown environment is reached, because the LSR contains the necessary information for the localization process, saving processing time while exploring.

Earlier works on EKF-SLAM consistency showed that an eventual inconsistency of the algorithm is inevitable for large-scale maps, and the estimated uncertainty will become optimistic when compared with true errors [7]. Some papers showed that the inconsistent estimate is necessary because of the errors introduced by the linearization process and can not be avoided. The violation of some fundamental constraints governing the relationship between various Jacobians, when they are evaluated at the current state estimate may cause inconsistency [8].

Some results can be detailed with an explicit analysis: 1) The measured noise is a factor that caused EKF-SLAM inconsistency; with an increase in the value of the noise, the degree of inconsistency decreased; with the increase in the observation times increase, the algorithm become inconsistent, but the inconsistency reached an upper bound; 2) The degree of improvement in the consistency with the increase of the measured noise is greater than the degree of damage to the consistency for the increasing in the observation time.

Figure 6 shows the odometry errors versus the obtained errors with the proposed approach. Determining the odometry errors of a mobile robot is very

important both in order to reduce them, and to know the accuracy of the state configuration estimated with the encoder data.

5 Conclusions and Future Work

Ever since SLAM was firstly introduced, many approaches have been investigated. Most of the early SLAM work was point-feature based. The main drawback with point-feature based SLAM is that measurements acquired from typical sensors did not correspond to point feature in the environment. After the raw sensor data is acquired, post processing is required to extract point features. This process may potentially introduce information loss and data association error. Further more, in some situation, the environment does not have enough significant structure to enable point features to be robustly extracted from them.

We can mention that we have developed a robust SLAM tool that is not limited to environments with linear features. The localization method is perfectly suited for new curves that can be increasingly seen in everyday's life. The theory and implementation of the B-splines was a powerful tool in our approach, and can be adapted to environments where the previous methods only considered simple descriptions.

As future work we have considered a great challenge: the extension of our proposal to the case of integrated exploration with multiple robots, which will take to us to the search of a solution to the multi-robot localization problem.

References

1. Makarenko, A.A., Williams, S.B., Bourgante, F., Durrant-Whyte, H.F.: An experiment in integrated exploration. In: IEEE Int. Conf. on Intelligent Robots and Systems, pp. 534–539 (2002)
2. Freda, L., Loiudice, F., Oriolo, G.: A randomized method for integrated exploration. In: IEEE Int. Conf. on Intelligent Robots and Systems, pp. 2457–2464 (2006)
3. Pedraza, L., Dissanayake, G., Valls Miro, J., Rodriguez-Losada, D., Matia, F.: Extending the limits of feature-based SLAM with B-Splines. IEEE Transactions on Robotics 25, 353–366 (2009)
4. de Boor, C.: A practical guide to splines. Applied Mathematical Sciences, vol. 27, Revised edn. Springer, Heidelberg (2001)
5. Rogers, D.F.: An introduction to NURBS with historical perspective. Morgan Kaufmann Publishers, San Francisco (2001)
6. Yamaguchi, F.: Curves and surfaces in computer aided geometric design. Springer, Heidelberg (1988)
7. Frese, U.: A discussion of simultaneous localization and mapping. Autonomous Robots 20(1), 25–42 (2006)
8. Huang, S., Dissanayake, G.: Convergence and consistency analysis for extended Kalman filter based SLAM. IEEE Transactions on Robotics 23(5), 1036–1049 (2007)

A Heuristic Approach for Constructing Ternary Covering Arrays Using Trinomial Coefficients

Jorge Martinez-Pena[1], Jose Torres-Jimenez[1],
Nelson Rangel-Valdez[1], and Himer Avila-George[2]

[1] CINVESTAV-Tamaulipas, Information Technology Laboratory
Km. 5.5 Carretera Cd. Victoria-Soto la Marina, 87130,
Cd. Victoria Tamps., Mexico
[2] Universidad Politécnica de Valencia - Instituto de Instrumentación para Imagen Molecular (I3M),
Camino de Vera s/n, 46022 Valencia, Spain
{jmartinez,jtj,nrangel}@tamps.cinvestav.mx, hiavgeo@posgrado.upv.es

Abstract. This paper presents a simulated annealing (SA) algorithm for the construction of ternary covering arrays (CAs) using a trinomial coefficient representation. A ternary CA, denoted by $\mathrm{CA}(t, k, 3)$, is an $N \times k$ array where each $N \times t$ subarray contains each of the 3^t combinations of symbols at least once. The construction of optimal CAs is, in general, an NP-complete problem. Many reported SA implementations use an $N \times k$ matrix representation for the CA construction. Instead of this, we represent ternary CAs using trinomial coefficients in order to reduce the search space for the SA algorithm.

Keywords: Ternary Covering Arrays, Trinomial Coefficients, Simulated Annealing, Software Testing.

1 Introduction

Testing a software system has become a mandatory activity for any software company that wants to accomplish international standards of quality [1]. It also serves as a kind of assurance to the customers that the final product will operate as they expected. However, considering the set of every parameter and their values for each module of the software system, the task to verify the interaction of the set of all parameters takes exponential time and hence it is impractical. Even when random test-suites are an option for the testing process, they do not offer any assurance of the functionality for the tested component. A technique to accomplish a specific level of functionality assurance in the testing process is called *software interaction testing*. This technique ensures that the interaction of every subset of t-parameters will be completely verified. Covering arrays (CAs) are combinatorial designs that are able to represent these interaction test-suites.

Sloane [2] defines a covering array as follows: a covering array (CA), denoted by $\mathrm{CA}(N; t, k, v)$, is an $N \times k$ array that has entries from the set $X = \{0, .., v-1\}$

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 572–581, 2010.

such that for every $N \times t$ subarray each of the v^t combinations are contained at least once. The parameter N is the number of rows, t is the *strength*, k the *degree* and v is the *order* or *alphabet*. When N is omitted, a CA can be denoted by $\mathrm{CA}(t, k, v)$.

We can construct an interaction test-suite directly from a CA. In this sense, the number of tests is the number of rows of the CA, the size of interaction is given by the CA strength, the number of parameters is the CA degree and the range of each parameter is represented by the CA alphabet. If the CA alphabet is set to three this combinatorial design is called ternary CA.

The most economical interaction test-suites will be formed if the used CA has been constructed with the minimum number of rows. The covering array number, denoted by $\mathrm{CAN}(t, k, v)$, represents the minimum value N for which a $\mathrm{CA}(t, k, v)$ can be constructed. Given a $\mathrm{CA}(N; t, k, v)$ with N equals to $\mathrm{CAN}(t, k, v)$, then it is called optimal [3].

The covering array construction (CAC) problem consists in obtaining optimal CAs. Even though the CAC problem is polynomially solved for some special cases [2,4], the general case is an NP-complete problem [5]. Therefore, many non exhaustive strategies have been applied to solve the CAC problem including heuristic approaches. For instance, Stardom [6] implemented a simulated annealing (SA) heuristic for the CAC problem. Torres-Jimenez [1] maintains a repository of CAs using as main construction tool a SA algorithm. Most of the CAs in this repository are the best known CAs in the literature.

Reported SA algorithms for the CAC problem usually employ an $N \times k$ matrix to represent the CA such that neighborhood functions are applied directly on that matrix. Besides the quality of the obtained CAs are frequently optimal or near optimal, the results of a heuristic algorithm depends directly on the input parameter value N for the algorithm, which most of the times it is unkown.

This paper presents a SA algorithm for constructing ternary CAs. This SA features the use of trinomial coefficient to represent the search space and does not require as input the parameter N to construct the ternary CAs. We propose a set of neighborhood functions that adequately explores and exploits the neighbors of a current feasible solution. It is well-known that the performance of a heuristic algorithm is sensitive to the parameter setting. In this sense, we included a methodology for fine tuning our SA implementation.

This research work continues as follows: Section 2 gives a panorama of the state-of-the-art for solving the CAC problem. Section 3 describes the trinomial coefficients and presents the key representation of ternary CAs by using this mathematical definition. Section 4 presents our heuristic approach, a SA algorithm for constructing ternary CAs that uses the trinomial coefficients to represent the search space. Section 5 details the computational experimentation. Finally, in Section 6 we give the conclusion of the results obtained from the experimentation.

[1] Torres-Jimenez. Covering Arrays Repository. Last accessed 3 August 2010. http://www.tamps.cinvestav.mx/~jtj/CA.php

2 Relevant Related Work

It is very important to obtain economical covering arrays (i.e. with the minimum number of rows). Although there are some cases where the solution is polynomial, most configurations of parameters t, k and v make the CAC problem an NP-complete problem. Due to the complexity of the CAC problem there are various non exhaustive strategies for solving the CAC problem. In this section we describe the relevant related work to the CAC problem.

In 1952, Bush [4] reported a direct method for constructing optimal CAs that uses Galois finite fields obtaining all $\mathrm{CA}(t, q+1, q)$ where q is a prime or a prime power and $q \leq t$. Another direct method is the zero-sum [7] that leads to optimal $\mathrm{CA}(t, t+1, v)$ for any $t > 2$. The method of zero-sum lists all v^t combinations, and concatenates horizontally a column that is the summation of the symbols for each combination modulo the alphabet.

In 2001, Stardom [6] made a study of different metaheuristics including Simulated Annealing (SA), Tabu Search (TS) and Genetic Algorithms (GA). He used a matrix of size $N \times k$ to represent the solution. His comparisons suggest that the SA algorithm was the best option to solve the CAC problem. Reported SA algorithms for the CAC problem usually employ a matrix to represent the CA such that neighborhood functions are applied directly to that matrix [6,8]. Besides the quality of the obtained CAs are frequently optimal or near optimal, the output result from a SA algorithm depends directly on the selected number of rows. This is, the SA algorithm needs the parameter N for the required CA to be searched. An extensive search must be performed for looking the best value for parameter N if that parameter is not set as an input parameter to the SA algorithm.

Nurmela [9] presented in 2004 a research work using a TS algorithm with a better design than the reported by Stardom. Nurmela initializes an $N \times k$ matrix by setting each cell randomly according to the alphabet. He uses the number of uncovered combinations as the evaluation function for a candidate solution. The TS tries to move in the neighborhood by selecting randomly any combination that is not covered yet. Then, it searches in the matrix where to make a single change trying to cover the missing combinations of symbols. The movement is taken if it is not tabu and the number of the total missing combinations is reduced. If a set of equal movement costs is presented, it selects a move randomly. This work reported new best known CAN for many instances.

In 2004, other strategies for obtaining ternary CAs were reported using recursive methods [10,11]. These algorithms make use of small CAs called ingredients in order to construct CAs with larger k. Many of the best known values of CAN for large k values are constructed using recursive methods.

Forbes et al. [12] in 2008, reported an algorithm for the efficient production of CAs of strength t up to 6 called IPOG-F. Instead of constructing one row at a time, this greedy method constructs test-suites one column at a time. This algorithm has been able to find some of the best known values of CAN for large k. However, it fails to obtain the optimal CAs constructed using direct methods [4]. Although the greedy approach has the guarantee of constructing a CA, the values of N are generally higher in comparison to other search strategies.

3 Trinomial Coefficients

Trinomial coefficients arise in the expansion of a trinomial (i.e. $(x+y+z)^k$). We use the trinomial coefficients as a key feature for the internal representation of our SA algorithm presented in Section 4. We give some important definitions of trinomial coefficients.

A trinomial coefficient [13], denoted by $\binom{k}{a,b,c}$, is a value that represents the number of combinations of length k by using three different symbols, say $0, 1$ and 2, where each symbol is used a, b and c times, respectively. Then, the integers a, b, c and k satisfy $(0 \leq a, b, c \leq k) \wedge (a + b + c = k)$. A trinomial coefficient is obtained by the Equation 1.

$$\binom{k}{a,b,c} = \frac{(a+b+c)!}{a!b!c!} \tag{1}$$

The trinomial coefficients are used in the trinomial theorem that enables the expansion of $(x + y + z)^k$. Graham et al. [14] define the trinomial theorem by the following identity,

$$(x+y+z)^k = \sum_{\substack{0 \leq a,b,c \leq k \\ a+b+c=k}} \binom{k}{a,b,c} x^a y^b z^c \tag{2}$$

Finally, the number of terms of the previous identity is $\binom{k+2}{2}$, and the summation of all trinomial coefficients is 3^k.

4 SA Algorithm for the Ternary CAC Problem

In this section we present a SA algorithm for constructing ternary CAs. See [15] for details of the SA heuristic. We describe each component of the implementation of our SA that uses a trinomial coefficient representation.

4.1 Internal Representation

We use the generated terms $\binom{k}{a,b,c}x^a y^b z^c$ of the trinomial theorem (from Equation 2) with a different meaning. We consider a term from the trinomial theorem as a *candidate row subset*, denoted by $\mathscr{R}^k_{a,b,c}$, as the row set formed by the expansion of a trinomial coefficient in terms of rows. In this sense, x, y and z represent the symbols $0, 1$ and 2, respectively, such that the term $x^a y^b z^c$ represents all the combinations formed by using x exactly a times, y exactly b times, and z exactly c times. Then, we know that the cardinality of $\mathscr{R}^k_{a,b,c}$ is $\binom{k}{a,b,c}$ and the number of columns in this row subset is k.

Let $\mathscr{T}$ be the set of all trinomial coefficients of degree k. Given $\mathscr{A} \subset \mathscr{T}$, the transformation of each element in the subset $\mathscr{A}$ into their respective *candidate row subsets* may lead to a ternary CA of a degree k. Moreover, if $\mathscr{A} = \mathscr{T}$, we automatically derive a CA$(3^k; k, k, 3)$. Therefore, it is possible to form a CA$(t, k, 3)$ for any $2 \leq t \leq k$ by using this representation.

In Table 1 we construct a ternary CA by using a subset $\mathscr{A}$ with cardinality four. The construction of a ternary CA using trinomial coefficients is done by transforming each trinomial coefficient in $\mathscr{A}$ to row subsets and then juxtaposing the row subsets.

Table 1. Representation of a CA$(9; 2, 3, 3)$ in trinomial coefficients. Left column describes a subset of trinomial coefficients while right column details the corresponding *candidate row subset* for each trinomial coefficient. A concatenation of these *candidate row subsets* forms the ternary CA.

$\mathscr{A}$	$\mathscr{R}^k_{a,b,c}$
$\binom{3}{3,0,0}$	0 0 0
$\binom{3}{0,3,0}$	1 1 1
$\binom{3}{0,0,3}$	2 2 2
$\binom{3}{1,1,1}$	0 1 2
	1 2 0
	2 0 1
	0 2 1
	1 0 2
	2 1 0

A feasible solution S in our SA algorithm is formed by the juxtaposing of some trinomial coefficients from $\mathscr{T}$. We implemented a binary selector vector to represent the juxtaposing of the trinomial coefficients for the solution. Each index $0 \leq i < \binom{n+2}{2}$ in S_i is an unique is an unique identifier to an element in $\mathscr{T}$. If S_i is turned on, it means that the i-th trinomial coefficient is selected for the solution. Finally, each time a value in S_i is changed, it is required a call to the evaluation function to correctly update S.

4.2 Initial Solution

We construct an initial solution by randomly selecting $t+k$ trinomial coefficients in S, where t is the strength and k the degree of the requested ternary CA. We empirically observe that $t + k$ trinomial coefficients were needed as an initial solution. Remark that when a single change in S is done, it is required to call the evaluation function. Then, the initial solution calls the evaluation function $t + k$ times.

4.3 Neighborhood Functions

Four neighborhood functions were implemented to guide the local search of our SA algorithm. These neighborhood functions are grouped in two different operators, *composition* and *formation*, and each operator acts in two different scopes,

random and *best-local*. Each neighborhood function is operates according to a probability of execution in $0.0 \leq p \leq 1.0$. Next we describe the neighborhood functions.

The *composition* operator travels into solutions that have different number of selected trinomial coefficients, i.e. solutions with different cardinality. This operator switches the value in the i-th position in S allowing to increase or to decrease the solution cardinality in terms of the number of selected trinomial coefficients. We propose two scopes for switching the value in a position at S. The *random* scope selects any position j in range $0 \leq j \leq \binom{k+2}{2}$ in a random fashion and switches the value in the j-th position. The *best-local* scope performs the $\binom{k+2}{2}$ switchings and selects the operation that minimizes the evaluation function. The cost of a single switch done by this operator is two calls to the evaluation function.

The *formation* operator travels into solutions that have different trinomial coefficients in a same cardinality solution. This operator exchanges the value in the i-th position for the value in the j-th position at S where $S_i \neq S_j$. Two different scopes for exchanging the values in S are given. The *random* scope randomly selects any position i where S_i is turned on and any position j where S_j is turned off. The *best-local* scope performs at most $\lfloor \binom{k+2}{2} \rfloor \lceil \binom{k+2}{2} \rceil$ exchanges and selects the operation that minimizes the evaluation function. The cost of a single exchange done by this operator is four calls to the evaluation function.

4.4 Evaluation Function

A neighbor of the current feasible solution is evaluated by the addition of two factors. The first factor refers to the number of uncovered combinations u. The second factor is the number N of rows of the neighbor divided by the total rows in our representation. The evaluation function to be minimized, denoted by δ, is described with the Equation 3.

$$\delta = u + \frac{N}{3^k} \tag{3}$$

This evaluation function was chosen for two reasons. The first reason is that when the SA algorithm finds a requested ternary CA the algorithm must not stop searching. Without the knowledge of the best value N in the representation of trinomial coefficients for the requested ternary CA, no function can evaluate if the optimal solution was found. More searching must be done in order to try to find a better ternary CA. The second reason is that the addition of the two factors can be interpreted as a two-objective search. The first objective is aimed to discover a ternary CA, i.e. no uncovered combinations are left in the ternary CA. The second objective refers to find a better ternary CA.

For determining the value u we implemented a local computation for performance improvement. This local calculation requires a matrix, called P, of size $\binom{k}{t} \times 3^t$. Each cell in P represents a single combination of each subarray for the requested ternary CA. Before starting the SA algorithm each cell in P is set to

zero. At any change in the current feasible solution S, we count each combination covered by the *candidate row subset* of the trinomial coefficient changed in S and update P. Any cell value in P greater than zero corresponds to a covered combination in S.

Thus, the total cost of a single call to an evaluation function is directly proportional to the trinomial coefficients currently being changed in S. Given any trinomial coefficient, the cost of performing this function is $\binom{k}{a,b,c}\binom{k}{t}$.

4.5 Termination Condition

The SA algorithm finishes with any of the following stop conditions: a) when the temperature is close to 1×10^{-4}; b) when the maximum number of iterations for the SA is reached; or c) if a number of calls to the evaluation function equals an established limit.

5 Computational Experiments

In this section we describe the experiments performed to evaluate the global performance of our SA algorithm presented in the previous section. This SA algorithm was implemented in C language and compiled with *gcc* using the optimization flag *-O3*. We run the experiments into a 3.4 GHz AMD Opteron and 4 GB of RAM Machine with a Linux operating system. Each of the experiments for benchmark comparison were executed 31 times using a maximum number of evaluation function calls to 1×10^6. For all the experiments, including the fine tuning methodology, the parameters of the SA were set as follows: the maximum number of iterations $I = 6000$, the markov chain length $L = 200$, the initial temperature $T = 1.0$, and the cold factor to $\alpha = 0.9$.

5.1 Fine Tuning the Probability of Execution of the Neighborhood Functions

It is well-known that the performance of a SA algorithm is sensitive to parameter tuning. In this sense, we follow a methodology for a fine tuning of the four neighborhood functions used in our SA algorithm.

The fine tuning was based on the next linear diophantine Equation,

$$p_0x_0 + p_1x_1 + p_2x_2 + p_3x_3 = q \tag{4}$$

In the previous Equation, x_i represents a neighborhood function and its value set to 1, p_i is a value in $\{0.0, 0.1, .., 1.0\}$ that represents the probability of executing x_i, and q is set to 1.0 which is the maximum probability of executing any x_i. A solution to the given linear diophantine Equation must satisfy the next Equation,

$$\sum_{i=0}^{3} p_i x_i = 1.0 \tag{5}$$

The Equation 4 has 286 solutions. Each solution is an experiment that test the *grade* of *participation* of each neighborhood function in our SA implementation to accomplish the construction of a ternary CA using the trinomial coefficients. Each experiment was run 5 times with a maximum number of evaluation function calls $E = 1 \times 10^4$.

The best 10 experiments from the 286 were run again 31 times each one with $E = 1 \times 10^6$. Finally, the analysis of the results leads to the configuration shown in Table 2. This configuration was selected to be used in the benchmark experimentation.

Table 2. Fine tuned parameters for the probability of execution p of each neighborhood function in our SA algorithm

Neighborhood function	p
Random composition	0.3
Best-local composition	0.4
Random formation	0.2
Best-Local formation	0.1

5.2 Benchmark Instances and Comparison Criteria

The benchmark instances used for comparing the results obtained with our SA algorithm are the test-suites obtained with the state-of-the-art IPOG-F [12] algorithm. The results were compared according to the value N of the ternary CAs constructed by both our SA and the IPOG-F. The values of strength were set to $2 \leq t \leq 6$ and the values of degree were set to $t+1 \leq k \leq t+4$. Results of IPOG-F algorithm were taken from the NIST Covering Array Tables webpage[2].

5.3 Benchmark Comparison

The results obtained with our algorithm show that the best values of N were given by our SA implementation than the IPOG-F algorithm for all the instances $\mathrm{CA}(t, t+1, 3)$ and $\mathrm{CA}(t, t+2, 3)$. However, for a degree $k \geq t+3$ the size of the $\mathrm{CA}(t, k, 3)$ instances obtained through IPOG-F were better (i.e. with less number of rows) than those obtained with our SA algorithm.

The benchmark comparison is shown in Table 3. The value of the degree for each instance is in function of the strength. We show the first four cases for all the instances $\mathrm{CA}(t, t+1, 3)$, $\mathrm{CA}(t, t+2, 3)$, $\mathrm{CA}(t, t+3, 3)$ and $\mathrm{CA}(t, t+4, 3)$. We can observe a trend in the results using our SA: the use of trinomial coefficients increments the number of rows of the constructed ternary CAs according to the strength. The instances of strength 2 present a linear growth, the instances of

[2] NIST Covering Array Table. Last accessed 3 August 2010. http://math.nist.gov/coveringarrays/ipof/ipof-results.html

Table 3. Comparison of the solution between our SA algorithm and the IPOG-F algorithm. t is the strength and k is the degree of a ternary CA$(t, k, 3)$. The degree is in function of the strength. Columns γ are values from our SA while columns β are values from the IPOG-F. Cases were SA improved IPOG-F results are bolted.

t / k	t+1		t+2		t+3		t+4	
	γ	β	γ	β	γ	β	γ	β
2	**9**	10	**12**	12	15	13	18	15
3	**27**	34	**40**	41	54	49	70	52
4	**81**	98	**126**	140	189	164	252	188
5	**243**	318	**378**	467	588	557	864	652
6	**729**	990	**1177**	1490	1968	1847	3060	2190

strength 3 and 4 present a quadratic growth, and the instances of strength 5 and 6 have a cubic growth. Moreover, the values of N for the instances CA$(t, t+1, 3)$ obtained with SA are optimal because $N = 3^t$ in all these cases.

6 Conclusion

In this paper we introduce a SA algorithm for the ternary CAC problem using trinomial coefficients as the representation of the ternary CAs. This algorithm implements the following key features:

- A novel representation of the search space using trinomial coefficients.
- A mixture of neighborhood functions. A set of four neighborhood functions were implemented. They were able to form the ternary CAs by exploring and exploiting diverse zones of the search space.
- An evaluation function that guides the search process. The evaluation function measures the number of missing combinations and the quality of the solution.

In order to provide a good global performance of the SA algorithm, we followed a fine tuning methodology for optimizing the assigned probabilities of execution for each of the four neighborhood functions using a linear diophantine equation.

Finally, we present results that improve the quality of some ternary CAs obtained with the IPOG-F algorithm. We could observe that the results for the instances CA$(t, t+1, 3)$ and CA$(t, t+2, 3)$ indicate that our SA implementation performs better than the IPOG-F results. Although our SA is prepared for working with higher strengths, we do not compare with cases where strength t is greater than 6 because IPOG-F does not report those results.

Acknowledgements. This research work was partially funded by the following projects: CONACyT 58554-Cálculo de Covering Arrays and 51623-Fondo Mixto CONACyT y Gobierno del Estado de Tamaulipas.

References

1. Hartman, A.: Software and hardware testing using combinatorial covering suites. In: Graph Theory, Combinatorics and Algorithms. Operations Research/Computer Science Interfaces Series, vol. 34, pp. 237–266. Springer US, Heidelberg (2005)
2. Sloane, N.: Covering arrays and intersecting codes. Journal of Combinatorial Designs 1, 51–63 (1993)
3. Ji, L., Yin, J.: Constructions of new orthogonal arrays and covering arrays of strength three. J. Comb. Theory, Ser. A 117, 236–247 (2010)
4. Bush, K.: Orthogonal arrays of index unity. Annals of Mathematical Statistics 23(3), 426–434 (1952)
5. Lei, Y., Tai, K.C.: In-parameter-order: A test generation strategy for pairwise testing. In: HASE, pp. 254–261 (1998)
6. Stardom, J.: Metaheuristics and the search for covering and packing arrays. Master's thesis, Simon Fraser University (2001)
7. Colbourn, C., Dinitz, J.: The CRC Handbook of Combinatorial Designs. CRC press, Boca Raton (1996)
8. Nurmela, K.J., Östergård, P.R.J.: Constructing covering designs by simulated annealing. Technical Report B10, Helsinki University of Technology, Digital Systems Labora tory, Espoo, Finland (1993)
9. Nurmela, K.J.: Upper bounds for covering arrays by tabu search. Discrete Appl. Math. 138, 143–152 (2004)
10. Martirosyan, S., van Trung, T.: On t-covering arrays. Des. Codes Cryptography 32, 323–339 (2004)
11. Hartman, A., Raskin, L.: Problems and algorithms for covering arrays. Discrete Mathematics 284, 149–156 (2004)
12. Forbes, M., Lawrence, J., Lei, Y., Kacker, R.N., Kuhn, D.R.: Refining the in-parameter-order strategy for constructing covering arrays. Journal of Research of the National Institute of Standards and Technology 213, 287–297 (2008)
13. Keeney, R.: On the trinomial coefficients. Mathematics Magazine 42, 210–212 (1969)
14. Graham, R.L., Knuth, D.E., Patashnik, O.: Concrete Mathematics: A Foundation for Computer Science. Addison-Wesley Longman Publishing Co., Inc., Boston (1994)
15. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science 220, 671–680 (1983)

Controlling Search in Constrained-Object Models

Ricardo Soto

Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso, Chile
ricardo.soto@ucv.cl

Abstract. A constrained-object model is a collection of classes parameterized by constraints and connected through composition and inheritance relations. A class is classically a factory of objects, that correctly linked are able to smoothly capture the inherent structure of complex and large constraint-based problems. Constrained-object languages are mainly focused on the definition of object-oriented, elegant, concise and expressive models. However, little attention has been paid in defining primitives to specify the search strategy of the solving process. In this paper, we address this concern by introducing an extensible formalism to define search strategies involving multiple options. Our main goal is to allow the user to customize, improve, and analyze the search process of constrained-object models.

Keywords: Constraint Satisfaction, Search.

1 Introduction

A Constraint Satisfaction Problem (CSP) is a formal problem representation that mainly consists of a sequence of variables lying in a domain and a set of constraints over such variables. Solving a CSP requires to find a variable-value assignment that satisfies the whole set of constraints. A constrained-object model is simply an object-oriented representation of a CSP. In this model, the classes encapsulate the variables and constraints of the problem, while typical relations such as composition and inheritance connect the classes to state a modular model. Indeed, a constrained-object model allows one to capture the compositional structure of a complex and large constraint-based problem in a more natural manner.

The resolution process of a CSP involves two main phases: modeling and search. In the modeling phase, the problem is analyzed and then encoded as a CSP in a given modeling or programming language. In the search phase, the model is launched in a solving engine, which attempts to find a result by performing backtracking-like procedures: a search tree is built by interleaving enumeration and propagation phases. The enumeration phase creates the branches of the tree by instantiating the variables and the propagation prunes it by eliminating those values that do not lead to any solution. In this process, different

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 582–591, 2010.

factors can impact the performance of the resolution, e.g., the order in which the variables and values are selected to create a branch as well as the chosen pruning mechanism.

The literature presents a considerable amount of work devoted to the design of constrained-object languages. In fact, one of the pioneers modeling languages for modeling CSPs was supported by an object-oriented framework [19]. There is also different constrained-object languages devoted to problems from the engineering field [3,11,9] as well as for typical benchmarks and well-known CSPs [16,18]. However, such languages are only focused on the modeling phase, lacking of a mechanism to specify the properties of the search process. Indeed, the solving engine is just a non-customizable black-box. This is a main drawback of constrained-object languages as the search strategy is the most important factor to get efficient resolution processes.

In this paper, we target this concern. We propose a simple and extensible formalism to specify the search properties. Value and variable ordering as well as consistency levels can be chosen by a simple parameter. This capability will make users able to tune, customize and/or improve the resolution process of constrained-object models. This work has been designed as part of the s-COMMA [4] system, a solver-independent architecture for modeling and solving CSP. In this framework, one s-COMMA model can be mapped through metamodeling techniques [5] to different solvers, for instance Gecode/J [1], ECLiPSe [20], RealPaver [10] and GNU Prolog [6].

It is worth mentioning that an important capability of the solver-independent architecture –also present in others state-of-the-art modeling languages [14,7,8]– is to allow users to test one model with different solvers. The goal is to experiment with different solving engines in order to determine which one performs better for a given model. Hence, by providing a customizable search process we clearly improve this capability, as the experimentation possibilities greatly increase. The user will be able to test multiple combinations of search options through different solving engines, having as a result, a wider vision of the solvers' behavior.

This paper is organized as follows: Section 2 gives an overview of CSPs and solving techniques. The s-COMMA language is introduced in Section 3. Section 4 presents the new formalism for specifying the value and variable ordering heuristics. The specification of pruning mechanisms is explained in Section 5. The extensibility of such a formalism is described in Section 6, followed by conclusions and future work.

2 Constraint Satisfaction Problems

Formally, a CSP $\mathcal{P}$ is defined by a triple $\mathcal{P} = \langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$ where:

- $\mathcal{X}$ is a n-tuple of variables $\mathcal{X} = \langle x_1, x_2, ..., x_n \rangle$.
- $\mathcal{D}$ is a corresponding n-tuple of domains $\mathcal{D} = \langle D_1, D_2, ..., D_n \rangle$ such that $x_i \in D_i$, and D_i is a set of values, for $i = 1, ..., n$.

– $\mathcal{C}$ is a m-tuple of constraints $\mathcal{C} = \langle C_1, C_2, ..., C_m \rangle$, and a constraint C_j is defined as a subset of the Cartesian product of domains $D_1 \times ... \times D_n$, for $j = 1, ..., m$.

Then, a constraint C_j is satisfied by a tuple of values $(a_1, ..., a_n)$ if $(a_1, ..., a_n) \in C_j$. The CSP is satisfied when all its constraints are satisfied. If the CSP has a solution, we say that it is consistent; otherwise we say that it is inconsistent.

2.1 CSP Solving

The CSP search phase is commonly tackled by building a tree structure by interleaving enumeration and propagation phases. In the enumeration phase, a variable and a value from its domain are chosen to create a tree branch. In the propagation phase, a consistency level is enforced to prune the tree, that is, the values that do not lead to any solution are temporarily deleted from domains. In this way, the exploration does not inspect unfeasible instantiations accelerating the whole process.

Figure 1 depicts a general algorithm for solving CSPs. The goal is to recursively generate partial solutions, backtracking when an inconsistency is detected, until a result is reached. The algorithm uses two data structures: *inst* and D. The former holds the instantiations while the latter the set of domains. The variable k represents the current level of the tree and *success* is a boolean variable to be set to true when a solution is found. The *instantiate* function is responsible for building the partial solutions and assigning them into the *inst* array. The *consistent* function decides whether the current instantiation can be extended to a full solution; additionally, it set *success* to true if the current instantiation is a solution. At the end, *restore* reinitializes the k variable's domain.

Let us notice that the value and variable selection are performed at line 2 and 10, respectively. The *propagate* procedure is responsible for pruning the tree by enforcing a consistency property on the constraints of the problem. The most used notion of consistency is the arc-consistency, which is defined in the following.

```
procedure solve(k : integer, inst : array)
1  while D[k] ≠ {} and not success do
2     a ← choose_value_from(D[k])
3     inst ← instantiate(inst, k, a)
4     if consistent(inst, k, success) then
5        if success then
6           print solution(inst)
7        else
8           propagate(k, D, failure)
9           if not failure then
10             l ← choose_variable()
11             solve(l, inst)
12 restore(k);
```

Fig. 1. Solving algorithm

Definition 1 (Arc Consistency). *Let C be a constraint over a set of variables $\{x_1, \ldots, x_n\}$ and let k be an integer, $k \in \{1, \ldots, n\}$. We say that C is arc consistent iff:*

$$\forall a_k \in D_k : \exists a_1 \in D_1, \ldots, \exists a_{k-1} \in D_{k-1}, \exists a_{k+1} \in D_{k+1}, \ldots, \exists a_n \in D_n \text{ such that } (a_1, \ldots, a_n) \in C$$

A constraint is said to be arc consistent if it is arc consistent wrt. to all its variables. A CSP is said to be arc consistent if all its constraints are arc consistent.

3 s-COMMA Language

s-COMMA is a strong-typed object-oriented modeling language for CSPs. Its core can be seen as a fusion of a classic object-oriented language and a constraint language. A s-COMMA model is a collection of classes connected through inheritance and composition relationships. A class is composed of attributes and constraints. An attribute is just a variable lying in a domain or an object. The constraints are stated as first-order logic formulas. Typical data structures such as arrays and control abstractions such as `forall` and conditional are also included. The main elements of the grammar are described below.

⟨*Model*⟩ ::= [⟨*Import*⟩]* [⟨*Class*⟩]*
⟨*Import*⟩ ::= **import** ⟨*Path*⟩
⟨*Class*⟩ ::= [**main**] **class** ⟨*Identifier*⟩ [**extends** ⟨*Identifier*⟩]
{⟨*Class-Body*⟩}
⟨*Class-Body*⟩ ::= [⟨*Attribute*⟩]* [⟨*Constraint-Zone*⟩]*

⟨*Attribute*⟩ ::= ⟨*Variable*⟩ | ⟨*Object*⟩
⟨*Variable*⟩ ::= ⟨*Var-Type*⟩ [**set**] ⟨*Mult-Id-Def*⟩ [**in** ⟨*Domain*⟩];
⟨*Mult-Id-Def*⟩ ::= ⟨*Identifier*⟩ [⟨*Array*⟩] [, ⟨*Identifier*⟩ [⟨*Array*⟩]]*
⟨*Object*⟩ ::= ⟨*Mult-Id-Def*⟩;
⟨*Var-Type*⟩ ::= ⟨*Basic-Type*⟩ | ⟨*Identifier*⟩
⟨*Basic-Type*⟩ ::= **int** | **real** | **bool**

⟨*Constraint-Zone*⟩ ::= **constraint** ⟨*Identifier*⟩ {⟨*Constraint-Body*⟩}
⟨*Constraint-Body*⟩ ::= [⟨*Constraint*⟩ | ⟨*Global-Constraint*⟩ | ⟨*Forall*⟩ | ⟨*If-Else*⟩]* [⟨*Optimization*⟩]
⟨*Constraint*⟩ ::= [[⟨*Cons-Level*⟩]] ⟨*Expression*⟩ ;
⟨*Literal*⟩ ::= ⟨*Value*⟩ | ⟨*String*⟩
⟨*Param*⟩ ::= ⟨*Access*⟩ | ⟨*Literal*⟩
⟨*Access*⟩ ::= [⟨*Identifier*⟩[⟨*Array*⟩].]*⟨*Identifier*⟩ [⟨*Array*⟩]

3.1 Real World Example

Now, as an example, let us present a s-COMMA model of a real-world CSP from Dassault Aviation about the design of an air conditioning system (ACS) for aircrafts. An ACS in an aircraft mixes hot air from the turbojet (main air) and cold air from atmosphere (ram air) to inject air at right temperature and pressure in the cabin (see Figure 2). The process begins once the main air is taken by the turbojet and sent to the compressor. This yields an increment of the pressure and temperature improving the heat transfers inside of the heat exchanger. Then, the main air flows through the turbine and its pressure and temperature

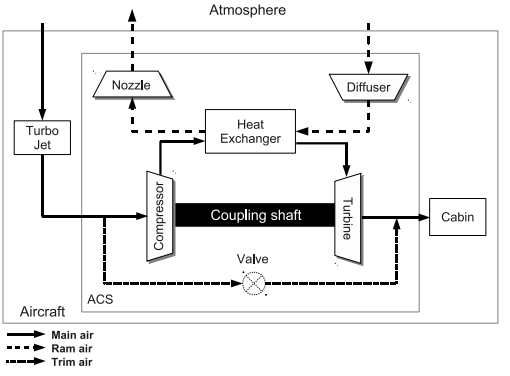


Fig. 2. The ACS topology

significantly decrease. The coupling shaft transfers the energy produced by the turbine rotation to the compressor. Finally, the valve regulates the air for an adequate cabin temperature.

In this problem, different constraints must be considered, mainly about the ACS design and temperature control. For instance, the air pressure and temperature of the cabin must be guaranteed for the well-being of the passengers. The entry air flows must be carefully managed, as they can decrease the turbojet performances and increase the aircraft drag force. Moreover, the volume and the mass of every element impact the aircraft performance. Hence, they must be minimized.

In the s-COMMA model for this problem (see Figure 3), the aircraft is composed of an air conditioning system `acs` and a turbojet `tj`, being instances of other classes. A constraint block called `Airflows` is defined to link the temperatures of air flows between the atmosphere, the turbojet, the ACS, and the cabin.

```
main class AirCraft {
  ACS acs;
  TurboJet tj;

  constraint Airflows {
    tj.TIn = TRam;
    tj.TOut = acs.THotIn;
    acs.THotOut = TCab;
    acs.TColdIn = TRam;
    ...
  }
}

 class HeatExchanger {
   real THotIn in [200,1000];
   real TColdIn in [200,1000];
   real THotOut in [200,1000];
   real TColdOut in [200,1000];
   ...
```

```
class ACS {
  HeatExchanger ex;
  Turbine turbine;
  Nozzle nozzle;
  compressor compressor;
  ...
  real THotIn  in [200,1000];
  real TColdIn in [200,1000];
  ...
  constraint AirflowsOne {
    ex.TColdOut = nozzle.TIn;
    THotIn = compressor.TIn;
  }
  ...
}
```

Fig. 3. An extract of the ACS model

A set of equality constraints is simply stated between air temperature attributes of `acs` and `tj`, and the flight conditions, namely the atmosphere air temperature `TRam` and the cabin air temperature `TCab`. These two parameters are defined as constants in a data file for given flight conditions. For the sake of simplicity, only a small extract of the full model is presented here. The other components are defined in the same way. In the rest of the paper we focus on the new formalism for specifying search processes.

4 Specifying the Variable and Value Ordering

The classical solving algorithm for CSPs explores the search space by trying to find a feasible instantiation i.e., a variable-value assignment that satisfies the whole set of constraints. The order in which the variables are selected to perform the instantiation refers to the variable ordering heuristic. Different research experiences [12,17] have demonstrated that a correct variable ordering can dramatically improve the performance of the search process. Currently, there exist several kind of variable ordering heuristics, for instance:

Name	Select the variable having:
min-dom-size	the smallest domain size.
max-dom-size	the largest domain size.
min-dom-val	the smallest value in its domain.
max-dom-val	the greatest value in its domain.
min-regret-min-dif	the smallest difference between the smallest value and the second-smallest value of its domain.
min-regret-max-dif	the greatest difference between the smallest value and the second-smallest value of its domain.
max-regret-min-dif	the smallest difference between the largest value and the second-largest value of its domain.
max-regret-max-dif	the greatest difference between the largest value and the second-largest value of its domain.

After selecting the variable to be instantiated, the value to be assigned must be selected. The order in which values are selected for the instantiation refers to the value ordering heuristic. Such a heuristic may also have a strong impact on the search process. For example, if the right value is chosen on the first try for each variable, a solution can be found without performing backtracks [12]. The most known value ordering heuristics are shown below.

Name	Select:
min-val	the smallest value.
med-val	the median value.
max-val	the maximal value.

In order to define the ordering heuristics in s-COMMA, it suffices to add the chosen parameter in the class header as follows:

⟨*Class*⟩ ::= [**main**] **class** ⟨*Identifier*⟩ [**extends** ⟨*Identifier*⟩] [`[`⟨*Ord-Heuristic*⟩`]`]

As an example, let us recall the ACS problem. It turns out that we have four possibilities. (1) To select the variable ordering, (2) to select the value ordering,

(3) to select both or (4) do not select any option, in this case the process will be launched using the default search provided by the solver. The four cases are shown in Figure 4.

```
// variable ordering selected
main class AirCraft [min-dom-size] {
  ...

// both selections
main class AirCraft [min-dom-size,min-val] {
  ...
```

```
// value ordering selected
main class AirCraft [min-val] {
  ...

// no selection
main class AirCraft {
  ...
```

Fig. 4. Four cases for selecting the ordering heuristics

Due to the searching process is performed for the entire problem, we cannot select different search options for each class. Hence, whether more than one class includes a search option, only the main class option will be considered.

5 Specifying the Consistency Technique

The use of an appropriate consistency technique can also improve the search process. Indeed, the *solve* procedure performs calls to the *propagate* function, which attempts to prune the tree in order to avoid the inspection of unfeasible space regions. There exist different propagation processes [12,17], most of them are based on the arc-consistency. We provide different consistency techniques:

Name	Select:
bound	An arc-consistency algorithm is used to reduce the domain of involved variables (only for discrete domains), but just the bounds of the variables' domain are updated .
domain	An arc-consistency algorithm is used to reduce the domain of involved variables (only for discrete domains). The full domain of variables is updated.
HC	A hull-consistency algorithm [13] is used to reduce the domain of involved variables. The hull-consistency can be seen as the arc-consistency for continuous domains.
BC3	A box-consistency algorithm [2] is used to reduce the domain of involved. Box-consistency handles the problem of multiple variable occurences when hull-consistency is used.
BC4	Combines the hull and the box consistency.

In order to define the consistency techniques we proceed analogously:

⟨*Class*⟩ ::= [**main**] **class** ⟨*Identifier*⟩ [**extends** ⟨*Identifier*⟩] [**[**⟨*Cons-Technique*⟩**]**]
⟨*Constraint*⟩ ::= [**[**⟨*Cons-Technique*⟩**]**] ⟨*Expr*⟩ **;**

Three cases are provided: (1) To select the consistency level for a class, (2) to select the consistency level for an object; and (3) to select the consistency level for a constraint. Let us note that cases 1 and 2 lead to a "cascade effect" i.e., the selected option will be inherited by objects and constraints belonging to the composition. Only objects and constraints with their own option do not inherit, they maintain their own selected option.

Two examples are shown in Figure 5. The class `AirCraft` has a `domain` consistency level. The "cascade effect" will set the objects and constraints of the Aircraft's composition (acs, tj, turbine, nozzle, etc) with the `domain` option, except for the `HeatExchanger` object `ex` and the constraint `ex.TColdOut = nozzle.TIn` which keep their own option (`BC4`). It is worth remarking that using this effect we profit of the object-oriented style to avoid defining the search option for constraints one by one.

```
// tuned class
main class AirCraft [domain] {
  ACS acs;
  TurboJet tj;
  constraint Airflows {
    tj.TIn = TRam;
    tj.TOut = acs.THotIn;
    acs.THotOut = TCab;
    acs.TColdIn = TRam;
    ...
  }
}
```

```
// tuned object & tuned constraint
class ACS {
  [BC4] HeatExchanger ex;
  Turbine turbine;
  ...
  constraint AirflowsOne {
    [BC4]  ex.TColdOut = nozzle.TIn;
    THotIn = compressor.TIn;
  }
```

Fig. 5. Specifying the consistency technique on the ACS problem

Let us note that the combination of consistency level with value and variable ordering is allowed.

```
class Aircraft [min-dom-size,min-val,HC] {
  ...
```

6 Extensibility

The solver-independent architecture is the state-of-the-art support of modeling languages for CSPs. This architecture consists of three layers: a upper layer holding the modeling language, a lower layer for the solver, and a middle layer for transforming the modeling language to the solver language. Among others, an interesting feature of this architecture is the capability of processing one model with different solvers. This feature is useful for experimentation tasks, considering that there exist many kind of models and there is no solver having the best resolution for all.

As a consequence, extensibility has became and important feature for modern architectures. For instance, consider that three new search options are introduced in the RealPaver solver. A variable ordering called `ROUND_ROBIN`, which selects the variables according to the lexicographic order. A value ordering called `BVAL_SPLIT_MIN`, which selects the first interval of the lower half of the domain; and the `BC3_NEWTON` consistency, which combines BC3 with the interval Newton Method [15].

One of the possible solutions for using those new search options from s-COMMA is to update its grammar and to recompile it by hand. Moreover, we need to redefine the RealPaver translator, i.e., the translation of the new options from the modeling language to the solver language must be encoded. In order to avoid

those hard tasks we introduce an extension mechanism to straightforwardly add new search options to the core language, that is, no grammar recompilation nor translator update is needed. It suffices to define an extension to specify the mapping process from s-COMMA to the solver language.

```
// extension file
RealPaver {
  Variable-Ordering {
    round-robin -> ROUND_ROBIN;
  }
  Value-Ordering {
    lower-half -> BVAL_SPLIT_MIN;
  }
  Consistency-Technique {
    BC3-newton -> BC3_NEWTON;
  }
}
```

```
// using the new options
class ACS [round-robin,lower-half] {
  [BC3-newton] HeatExchanger ex;
  Turbine turbine;
  ...
  constraint AirflowsOne {
    [BC3-newton] ex.TColdOut = nozzle.TIn;
    THotIn = compressor.TIn;
  }
}
```

Fig. 6. Extension for RealPaver and the custom `ACS` class

An extension file is depicted in Figure 6. It may be composed of one or more main blocks. A main block defines the solver where the new functionalities will be defined. Indeed, extensions for more than one solver can be defined in the same file. Within a main block, we define the blocks for the search options: a `Variable-Ordering` block, a `Value-Ordering` block, and a `Consistency-Technique` block. Inside each block we define the rules of the translation. The grammar of the rule is as follows:

$$\langle \textit{s-COMMA-option} \rangle \rightarrow \langle \textit{solver-option} \rangle \ ;$$

On the translation process the $\langle \textit{s-COMMA-option} \rangle$ will be translated to the $\langle \textit{solver-option} \rangle$. Figure 6 shows the use of this new options. The variable ordering `ROUND_ROBIN` can be chosen using the parameter `round-robin`. The value ordering `BVAL_SPLIT_MIN` can be selected using `lower-half`; and the `BC3_NEWTON` consistency technique is used by means of `BC3-newton`.

7 Conclusion and Future Work

In this paper, we have presented a simple and extensible formalism to specify search options in constrained-object models. This new approach allows one to specify multiple combinations of value and variable ordering heuristics along with consistency techniques. Consistency techniques can be smartly defined at different levels of the model: class consistency technique, object consistency technique and constraint consistency technique. In this stage, a simple "cascade effect" allows one to avoid rework by automatically transmitting the chosen option through the composition hierarchy.

The extension mechanism introduced is an important feature of the system. It is possible to include new search options within the core language by defining a simple extension file. This capability make the s-COMMA architecture adaptable to further up-grades of the solvers.

The formalism introduced here is ongoing work, and we believe there is a considerable scope for future work, for instance, the addition of search-tree exploration options, or the addition of search patterns for optimization and local search.

References

1. Gecode System, http://www.gecode.org (visited 6/2010)
2. Benhamou, F., Mc Allester, D., Van Hentenryck, P.: CLP(Intervals) Revisited. In: Proceedings of ILPS, pp. 124–138. MIT Press, Cambridge (1994)
3. Borning, A.H.: The Programming Languages Aspects of ThingLab, a Constraint-Oriented Simulation Laboratory. ACM TOPLAS 3(4), 353–387 (1981)
4. Chenouard, R., Granvilliers, L., Soto, R.: Model-Driven Constraint Programming. In: Proceedings of ACM SIGPLAN PPDP, pp. 236–246. ACM Press, New York (2008)
5. Chenouard, R., Granvilliers, L., Soto, R.: Rewriting Constraint Models with Metamodels. In: Proceedings of SARA, pp. 42–49. AAAI Press, Menlo Park (2009)
6. Diaz, D., Codognet, P.: The GNU Prolog System and its Implementation. In: Proceedings of SAC (2), pp. 728–732 (2000)
7. Nethercote, N., et al.: Minizinc: Towards a standard cp modelling language. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 529–543. Springer, Heidelberg (2007) (to appear)
8. Frisch, A.M., Grum, M., Jefferson, C., Martínez-Hernández, B., Miguel, I.: The design of essence: A constraint language for specifying combinatorial problems. In: Proceedings of IJCAI, pp. 80–87 (2007)
9. Fritzson, P., Engelson, V.: Modelica – A Unified Object-Oriented Language for System Modeling and Simulation. In: Jul, E. (ed.) ECOOP 1998. LNCS, vol. 1445, p. 67. Springer, Heidelberg (1998)
10. Granvilliers, L., Benhamou, F.: Algorithm 852: RealPaver: An Interval Solver Using Constraint Satisfaction Techniques. ACM TOMS 32(1), 138–156 (2006)
11. Jayaraman, B., Tambay, P.Y.: Modeling Engineering Structures with Constrained Objects. In: Krishnamurthi, S., Ramakrishnan, C.R. (eds.) PADL 2002. LNCS, vol. 2257, pp. 28–46. Springer, Heidelberg (2002)
12. Apt, K.R.: Principles of Constraint Programming. Cambridge Press, Cambridge (2003)
13. Lhomme, O.: Consistency Techniques for Numeric CSPs. In: Proceedings of IJCAI, pp. 232–238 (1993)
14. Marriott, K., Nethercote, N., Rafeh, R., Stuckey, P.J., de la Banda, M.G., Wallace, M.: The design of the zinc modelling language. Constraints 13(3), 229–267 (2008)
15. Neumaier, A.: Interval Methods for Systems of Equations. Cambridge University Press, Cambridge (1990)
16. Paltrinieri, M.: A Visual Constraint-Programming Environment. In: Montanari, U., Rossi, F. (eds.) CP 1995. LNCS, vol. 976, pp. 499–514. Springer, Heidelberg (1995)
17. Dechter, R.: Constraint Processing. Elsevier, Amsterdam (2003)
18. Soto, R., Granvilliers, L.: The Design of COMMA: An Extensible Framework for Mapping Constrained Objects to Native Solver Models. In: Proceedings of IEEE ICTAI, pp. 243–250 (2007)
19. Sutherland, I.E.: Sketchpad, A Man-Machine Graphical Communication System. In: Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York (1963)
20. Wallace, M., Novello, S., Schimpf, J.: ECLiPSe: A Platform for Constraint Logic Programming. Technical report, IC-Parc, Imperial College, London (1997)

Adaptivity and Interoperability in e-Learning Using Ontologies

José dos Reis Mota and Márcia Aparecida Fernandes

Universidade Federal de Uberlândia,
Uberlândia, Minas Gerais, Brasil
josereis@unipam.edu.br, marcia@ufu.br
http://www.facom.ufu.br/posgrad/

Abstract. Adaptive systems in the field of e-learning are those capable of adapting the content and mode of presentation according to the profile of a particular user or user group. These systems generally remain in the context of academic research and are rarely used in real situations. Moreover, the e-learning environments that present a complete platform of tools and resources available are largely used by educational institutions, but are not adaptive. We propose to integrate these two types of systems, aiming to take advantage of both - adaptivity coupled with the scope of use of e-learning environments. Therefore, we describe the implementation of an agent that integrates an intelligent adaptive system with the e-learning environment Moodle. Furthermore, we design and implement an adaptive system based on ontologies and e-learning standards. We hope to contribute to making e-learning environments best suited to provide meaningful learning, as well as to facilitate the creation of courses.

Keywords: adaptivity, e-learning, interoperability, ontology, semantic web.

1 Introduction

Several studies emphasize the need for development of adaptive educational systems, which fit the profile of each student, and interoperable, enabling the sharing of resources between systems. Such characteristics are important in e-learning environments, as the ability to meet the overwhelming demand from students, and thus it is necessary to suit different user profiles and learning contexts. Currently, there are several available e-learning environments that are used by educational institutions, including some open source options such as Moodle [1], Teleduc [2] and Claroline [3]. Such systems have various resources and tools such as forums, chats, authoring and evaluation tools. However, they are still restricted in relation to adaptivity.

Moreover, the existing adaptive systems remain mostly in the context of academic research and are not applied to real situations. One reason is that, by proposing a system enabling adaptivity, the researchers do not usually worry

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 592–601, 2010.

about integration with other systems, and implement a complete platform for e-learning, with all the necessary resources. Thus, institutions that already offer courses taking advantage of this feature would have to migrate to a new system, which in most cases is not feasible due to cost and time.

The Artificial Intelligence techniques have been widely used in proposed intelligent adaptive systems, such as Planning and Multiagent Systems [4][5]. In fact, in order to realize adaptivity it is necessary to consider a number of characteristics of the students and of the content, and, therefore, rules are needed to combine these features, and actions can be implemented as a result of these rules.

The semantic web technologies have also been used as an alternative to allow adaptivity and interoperability in e-learning systems, because it is an approach that seeks to overcome limitations of the current web in order to promote interoperability, enabling different applications to share resources. It also allows the implementation of intelligent agents, capable of allowing adaptability, using search engines and inference, as described in [6].

This paper proposes an adaptive system based on ontologies that can be integrated with e-learning environments. And it describes the integration process of this proposal to Moodle, an e-learning environment widely used.

This paper is organized as follows: Section 2 presents the background konwledge, taking into account mainly the components related to adaptivity. In Section 3, the ontologies that compose the adaptive system are defined. Section 4 describes the integration process of the adaptive system proposed to Moodle. Finally, section 5 presents the conclusions and future works.

2 Background Knowledge

Adaptive systems in the field of e-learning are those capable of adapting the content and mode of presentation according to the profile of a particular user or group of users. Thus, users with different learning goals, preferences and knowledge can access different contents with different presentation formats. Brusilovski and Peylo [5] classify adaptive and intelligent web-based educational systems as those that "attempt to be more adaptive by building a model of the goals, preferences and knowledge of each individual student and using this model throughout the interaction with the student in order to adapt to the needs of that student. They also attempt to be more intelligent by incorporating and performing some activities traditionally executed by a human teacher".

Also according to Brusilovski and Peylo [5], the two major technologies exploited by the systems of adaptive hypertext and hypermedia are Adaptive Presentation and Adaptive Navigation Support. The goal of the Adaptive Technology Presentation is to adapt the content present in each hypermedia page according to the goals of the student, their knowledge, and other information stored in the student model. The Adaptive Navigation Support aims to help students in orientation and navigation by changing the appearance of visible links. For example, a hypermedia system can sort, enhance or partly hide the

links on the current page to make it easier to choose the next step. The adaptive navigation helps students find an "optimal path" for the learning material, as well as the curriculum sequencing.

Wu [9] shows that adaptive systems, conceptually, consist of three basic components that are not necessarily represented separately. The first component is the domain model that represents the content displayed by the system and the relationships between them. The second one is the user model that contains the relevant features for adaptation, such as preferences, knowledge, goals, and browsing history. The third one is the adaptation model, which has updated the user profile, according to the observation of their behavior, and determines how to accomplish the adaptation. Thus, this latter component uses information from the other components.

According to this conceptual view, a classic model of adaptation must contain a domain model, a student model and rules of adaptation. ActiveMath [10] [11], AHA! (Adaptive Hypermedia for All) [12] [13] and Simeduc [4] are examples of adaptive systems containing these components.

ActiveMath is a system dedicated to the teaching of mathematics. The representation of the content is based on an XML (eXtensible Markup Language) [14]. For each user, the appropriate content is retrieved from a knowledge base and the course is generated individually according to pedagogical rules, objectives, user preferences, skills and learning context.

The architecture of the ActiveMath System, described in Fig. 1, consists of primary components, which are used for system configuration, system management and exchange of information between User, Events, Web Core and Layer, and tightly coupled components, which are the central part of the application and may be associated with the basic components of an adaptive system.

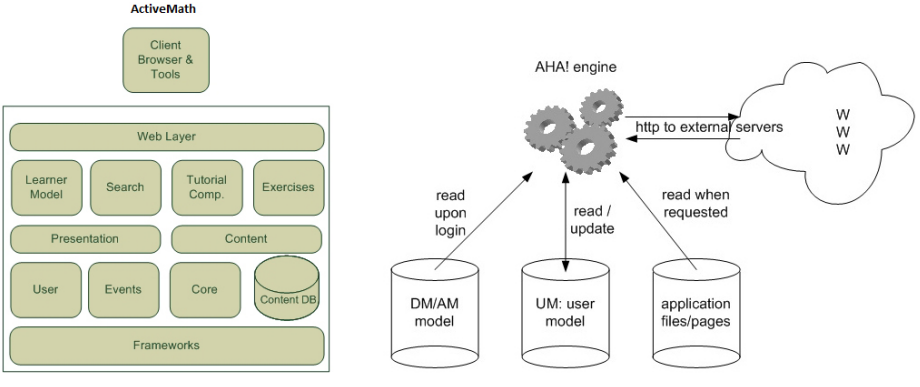


Fig. 1. Architecture of ActiveMath and AHA Systems

AHA! (Adaptive Hypermedia for All) [12] [13] is also an example of adaptive system. This is an open source software. The student model is defined by concepts that contain attributes. The adaptability is determined by relationships

or prerequisite conditions defined by a boolean query. Fig. 1 shows the main components of this system.

AHA!Engine can use local information and, potentially, can also use resident pages on other servers on the web. The domain and adaptation model (DM / PM) represents a model of conceptual structure and rules of adaptation. This information is retrieved when the user accesses the system. When the user interacts with the application, the component user model (UM) is used and constantly updated. DM / AM and UM are used to decide which pages or files should be retrieved in any specific request of the user, and which objects should be included in a conditional way on the page.

Regarding these adaptive systems, it was observed that both ActiveMath and AHA! have the basic components mentioned by [9], but do not present functionalities to cope with reusability, since the courses are not created according to appropriate standards for this purpose.

The ontologies have been used as an alternative to provide adaptivity to the e-learning environments, as they are able to capture the knowledge of a specific area, enabling the definition, formalization and sharing concepts, constraints, instances, axioms and relationships in that area. The work developed by [15] and [16] are examples of systems that use ontologies and metadata associated with learning objects to promote adaptivity, specifically the IEEE LOM metadata.

Munoz [15] proposes a system based on semantic web technologies to provide adaptivity, and achieves interoperability in e-learning linking the existing metadata standards for e-learning with the resources of the Semantic Web (RDF / XML), and modeling ontologies needed for the proposed architecture. However, this work does not include, more broadly, the issue of adaptivity, since it does not populate the student ontology or implement capable agents to use it.

In Branco Neto [16], the ontologies include information about the knowledge domain, learning theories and students. The adaptivity is a preliminary step, when creating the course. Subsequently, courses tailored and based on the SCORM standard can be imported into e-learning environments that support the standard.

The analysis of adaptive systems and proposals based on semantic web technologies allowed to identify the necessary information that should be extracted from a Learning Management System (LMS), in order to design and develop an adaptability process. Then, this work proposes to develop an adaptive system based on ontologies that can be integrated into e-learning environments to provide adaptivity and achieve interoperability. In consequence, an agent to perform this integration has been implemented.

3 Adaptive System Based on Ontologies

Based on Wu [9], who defines an adaptive system for e-learning environments and semantic web technologies, this work proposes three ontologies that correspond to the three basic components to achieve adaptation: the LOM ontology for representing the learning objects on the IEEE-LOM standard; the domain

ontology, which has the main concepts associated with the course structures in e-learning environments; and the pedagogical ontology, which links the two previous ontologies and defines the aspects related to the student profile.

The LOM Ontology is based on the IEEE-LOM that is used as a benchmark by the most systems to describe learning objects, including the SCORM standard. In this work the course structure through the content modules has been adopted. Some projects, such as [17] have already established a representation in RDF (Resource Description Framework) for the LOM standard. However, in this study were defined only the standard items which were necessary for processing the adaptation rules. Then, three specifications of IEEE-LOM were created: general, classification and educational. In the case of the educational specification, the sub-elements were considered: interactivity type, semantic density, difficulty level and learning resource type. Each learning object noted in SCORM packages of a course is mapped as an instance of this ontology.

In the Domain Ontology, the concepts that compose the course content and their relations are described. Due to the adopted standard for the course construction, it is possible to share this ontology among different systems.

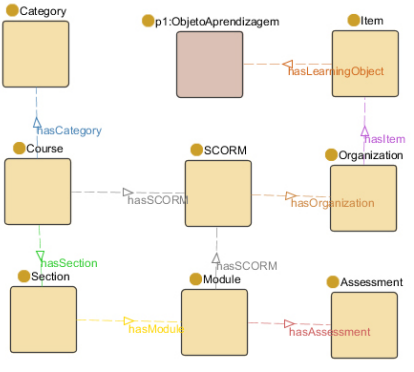


Fig. 2. Classes and Object Properties of the Domain Ontology

As it is shown in Fig. 2, a course consists of sections, which in turn may contain several modules. It was assumed that the contents of a module should be structured in a SCORM package, due to the benefits already mentioned. Besides, each module must have activities for the student evaluation. A SCORM package may have several organizations, which correspond to the structure of a course at a certain level of complexity. In each organization there are the items related to the content topics, and each item can be associated with several learning objects which are already modeled in the LOM ontology. This allows the use of adaptation rules that take into account the course complexity, as well as the level of learning objects, by enabling the selection of those most appropriate to the profile of each student.

Finally, the Pedagogical Ontology was defined, as one can see in Fig. 3, which describes the relations between the LOM and domain ontologies. Another important aspect of this ontology is the ability to consider the learning styles in the pedagogical actions and adaptation rules. According this theory, proposed by Felder [18] [19], the learning styles are divided into four dimensions of the learning process: perception, acquisition, processing and understanding. So, each student has a percentage of each dimension and it is possible to identify the predominant learning style.

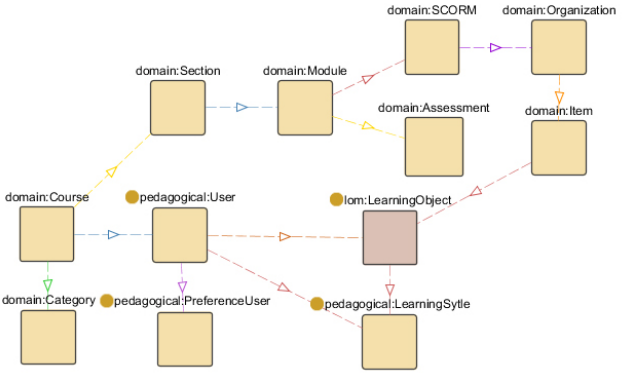


Fig. 3. Classes and Relations of Pedagogical Ontology

The ontology modeling made use of the Protege tool [20], which supports the creation of ontologies in OWL (Ontology Web Language) [21]. It also allows the graphical representation of elements and the use of reasoners to check the consistency of the ontologies, and to infer classes which were not explicitly presented in the representation.

The generation of instances for the ontologies was done using Jena [22], which is a Java framework for building semantic web application and supporting queries to ontologies and inference mechanisms, by API means that manipulate RDF and OWL. The ontology LOM was populated by learning objects described in the SCORM packages. In the Moodle database there is no reference to metadata objects. However, all the files present in the SCORM packages are stored in the data folder system. Thus, the execution of the file denominated imsmanifest.xml can generate instances for each learning object.

The data that populate the domain ontology are found directly in the Moodle database. Then, only the mapping of the values to the properties present in the ontology was necessary. Also in the pedagogical ontology, part of the data related to students can be found in the database. But the student model of Moodle has been extended to include data about learning styles, which according to Felder [20] can be obtained through the questionnaire available at [23].

3.1 Adaptation Rules

As it was mentioned in the previous sections, an adaptive system should be able to distinguish the student and select the most appropriated learning object that considers the student profile and pedagogical aspects. In this proposal, three specific aspects are highlighted: the evaluation process that considers the student behavior; the course formation that allows different organizations due to the use of SCORM standard; and the learning styles that describes the student on the pedagogical point of view.

Each topic in Moodle - called learning section - contains a SCORM package and other activities available in the environment. Initially, besides the present activities in the SCORM packages, the following activities were also considered: questionnaires for assessment of student knowledge in relation to the learning section; forum and chat that are associated with the student behavior during the course.

The evaluation process is based on Faria et al. [24], and takes into account the behavior and the knowledge of the student. The behavior evaluation considers the spent time to finish the learning of all contents, as well as the spent time to conduct the evaluation, forum and chat participations. The knowledge evaluation considers the percentage of correct answers. These values are combined into fuzzy rules that give a feedback to the student performance. This result and the student profile that determined the next organization will be followed.

After enrolling in the course, the student answers the questionnaire [23], that identifies the learning styles according to the scale set and to the dimensions presented in [18]. As each package provides several organizations in an increasing degree of complexity, in the first moment the system will provide to the student an organization of intermediate complexity taking into account the learning style associated with the student.

The adaptation process starts from the learning style of the student. Then, the adaptive system searches for instances of LOM ontology related to this learning style based on the annotation of the learning resource type. In the next step, it is verified which of these learning objects have already been presented to the student, and the system selects those that the student did not know yet. If any instance is found, the system will consider the student performance in the learning section.

So, in the last step, the system uses the IEEE LOM specifications associated with the learning objects in order to find those that are most appropriate - firstly, in relation to the degree of difficulty and, after, in relation to semantic density. If the student performance is lower than the grade set as a parameter, the student must review the same learning section with an organization of lower complexity. On the other hand, if the performance is equivalent to or higher than the grade set, the system will provide a new section with an organization of intermediate complexity or greater complexity.

4 Integrating Adaptive Systems to LMS-Moodle

Initially, the integration of Moodle with the agent platform of Simeduc was made, which is an intelligent adaptive system, consisting of four agents: expert, evaluation, pedagogical and assistant. Besides, the evaluation system of Simeduc is based on Fuzzy Logic and neural network, which considers the results of the learning content by the answers of the questionnaires for each leaning section, and by the behavior which is based on the access to the communication, such as forum and chat. SIMEDUC does not use learning styles in the student model.

This first integration has demonstrated the feasibility of integrating the LMS adaptive systems. Besides, this integration process allows designing and developing an agent able to link the two environments easily and efficiently. Thus, this same agent was used in the integration of the adaptive system based on ontologies proposed in this work.

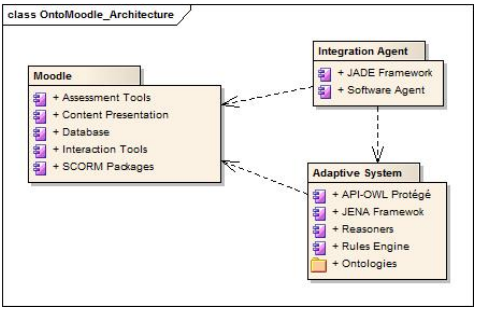


Fig. 4. Architecture of Adaptive Moodle

A detailed study of the Moodle platform allowed finding the existing information that could populate ontologies and the integration points to make the bridge between Moodle and the adaptive system. The most important information was the data base, as well as the content presentation, assessment tools, communication tools and SCORM packages. Fig. 4 shows the architecture of the system proposed, where one can see the integration agent that is responsible for the integration by receiving the information about the student (chat and forum participations, assessment, profile and time spent with activities), and for sending them to the adaptive system, so that the adaptation rules can be applied and the new pedagogical actions can be proposed.

One important characteristic of Moodle is the capability of defining one course in various formats. This is useful for an adaptive system because it allows the course customization according to each student profile and learning goals. The most common format is topical or weekly. Each topic can be associated with resources or activities. A SCORM package can be the only activity that contains one or more contents, or a complete course with all the contents.

Among the technical requirements for implementing a system that integrates an adaptive system to an e-learning environment, the two most important requirements considered in this project were the development of an environment-independent platform and an easy approach of integration. This way, there would be a weak coupling with the features of the systems involved, so that stakeholders may choose to use resources to adapt to certain courses, and keep intact the structure of the courses already entered.

Thus, from the information gathered from the analysis of the Moodle environment, the integration agent using the technology of multi-agent systems and the framework JADE (Java Agent Development Framework) [25] was designed and developed. The agent automatically starts to monitor the integration by taking Moodle data in relation to the new students that are enrolled in the courses. If the course uses the adaptation process, the integration agent puts both systems together - adaptive and e-learning environment. At the beginning of the integration process a copy of the course for each user is generated. This course shares the activities of the original course; however, the contents provided by the SCORM packages are specific to each student, as defined in the adaptation rules.

5 Conclusions

This paper has demonstrated that the integration of adaptive systems to an e-learning environment is feasible. As example two adaptive systems were integrated to Moodle. The first one is an intelligent multiagent system, and the second one is based on the ontologies. In order to achieve these integrations, an integration agent was developed, which can be handled for integrating an adaptation to other e-learning environments.

We observe that the two systems in each integration remain independent, and it is worthwhile. Consequently, this characteristic allows the creation of adaptive courses without the necessity of migration from an environment that is already being used. Then, the e-learning environments can be more appropriated to provide meaningful learning and consider the specific needs of each student.

We also presented the design and development of an adaptive system based on ontology that takes into account information generally present into the e-learning environment, but mainly information about the learning styles and metadata in order to propose the adaptation rules. Then, these two aspects made up an innovative system in relation to adaptation.

In addition, an important point to observe is the capability to extend the student model, so that information that is not present in the e-learning environment can be used by the adaptation process. This can be noted in relation to the learning styles of the students.

References

1. Moodle - Learning Management System, `http://moodle.org`
2. Teleduc, `http://teleduc.nied.unicamp.br`

3. Claroline.NET, http://www.claroline.org
4. Dorca, F.A.: An intelligent multiagent system for distance education. Master's Thesis. Graduate Program in Computer Science. Federal University of Uberlandia (2004)
5. Brusilovsky, P., Peylo, C.: Adaptive and intelligent web-based educational systems. International Journal of Artificial Intelligence in Education 13, 156–169 (2003)
6. Dolog, P., Nejdl, W.: Semantic web technologies for the adaptive web. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 697–719. Springer, Heidelberg (2007)
7. SCORM - sharable content object reference model, http://www.adlnet.gov/
8. IEEE LOM - IEEE learning object metadata, http://ltsc.ieee.org/wg12/index.html
9. Wu, H.: A reference architecture for adaptive hypermedia applications. Eindhoven: technische Universiteit Eindhoven, Doctoral Thesis (2002), http://alexandria.tue.nl/extra2/200213488.pdf
10. Melis, E., Andres, E., Büdenbender, J., Frishauf, A., Goguadse, G., Libbrecht, P., Pollet, M., Ullrich, C.: ActiveMath: A web-based learning environment. International Journal of Artificial Intelligence in Education 12(4), 385–407 (2001)
11. ActiveMath, http://www.activemath.org/
12. De Bra, P., Smits, D., Stash, N.: Creating and Delivering Adaptive Courses with AHA! In: Nejdl, W., Tochtermann, K. (eds.) EC-TEL 2006. LNCS, vol. 4227, pp. 21–33. Springer, Heidelberg (2006), http://wwwis.win.tue.nl/~debra/ec-tel2006.pdf
13. AHA! Adaptive Hypermedia for All, http://aha.win.tue.nl/
14. XML - eXtensible Markup Language, http://www.w3.org/XML
15. Munoz, L.S.: Ontology-based Metadata for e-learning Content. Master's Thesis, Graduate Program in Computer Science. Federal University of Rio Grande do Sul, Porto Alegre (2004)
16. Branco Neto, W.C.: Semantic Web in the construction of adaptive learning systems. Doctoral Thesis. Graduate Program in Computer Science. Federal University of Santa Catarina (2006)
17. Nilsson, M., Palmr, M., Brase, J.: The LOM RDF binding - principles and implementation. In: Proceedings of the 3rd Annual ARIADNE Conference, Katholieke Universiteit Leuven, Belgium (2003)
18. Felder, R.M., Henriques, E.R.: Learning and Teaching Styles in Foreign and Second Language Education. Foreign Language Annals 28(1), 21–31 (1995), http://www.ncsu.edu/felder-public/Papers/FLAnnals.pdf
19. Felder, R.M., Silverman, L.K.: Learning and Teaching Styles in Engineering Education. Engr. Education 78(7), 674–681 (1988), http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/LS-1988.pdf
20. Protege, http://protege.stanford.edu/
21. W3C: OWL Web Ontology Language Reference (2004), http://www.w3.org/TR/owl-ref/
22. Jena - a semantic web framework for Java, http://jena.sourceforge.net/
23. Felder, R.M., Soloman, B.A.: Index of Learning Styles Questionnaire, http://www.engr.ncsu.edu/learningstyles/ilsweb.html
24. Faria, M.N., Malva, G.R.O., Dorca, F.A., Lopes, R.S., Fernandes, M.A.: A assessment system based on fuzzy logic. Horizonte Cientfico 1(8) (2008), http://www.seer.ufu.br/index.php/horizontecientifico/article/view/3971
25. JADE - Java Agent DEvelopment Framework, http://jade.tilab.com/

Author Index

Abundez Barrera, Itzel 144
Alsinet, Teresa 40
Aluisio, Sandra 306
Alvarado-González, Montserrat 512
Arregi, O. 234
Arroyo-Figueroa, Gustavo 473
Arroyo, José E.C. 70
Ávila, Bráulio C. 443
Avila-George, Himer 30, 572
Avilés, Héctor 10, 512

Baffa, Augusto C.E. 110
Barbosa-León, Hector 1
Barrón-Estrada, M.L. 1
Bayá, Ariel E. 184
Béjar, Ramón 40
Bernardini, César 503
Billhardt, Holger 274
Brasil, Christiane R.S. 50

Cabanillas, Javier 552
Cardenas, Marina E. 366
Carvalho, Ariadne Maria Brito Rizzoni 336, 376, 386
Castellanos, Hayde 346
Castillo, Julio J. 366
Ceberio, K. 234
Chavarría, Alejandra 346
Ciarlini, Angelo E.M. 110, 133
Coelho, Helder 254
Cohen, Andrea 90

da Cruz Jr., Gelson 296
da Rocha Costa, António Carlos 254
da Silva, Adenilton J. 396
da Silva, Fernando José Vieira 336
De Bona, Glauber 453
de Carvalho, André C.P.L.F. 194
De Giusti, Armando 60
Delgadillo, Miguel Angel 463
de Novais, Eder Miranda 316, 326
de Souza, Bruno Feres 194
Dias, Maurício A. 542
Dias Tadeu, Thiago 316, 326
Díaz de Illarraza, A. 234
Díaz Zagal, Sergio 144
Domínguez, Enrique 406
dos Reis Mota, José 592
dos Santos, Paula M. 70

Enembreck, Fabrício 443
Espinoza, Judith 532

Falappa, Marcelo A. 100, 356
Fernandes, Leandro C. 542
Fernandes, Márcia Aparecida 592
Finger, Marcelo 453

Garcia, Alejandro J. 90, 284
Gasperin, Caroline 306
Godo, Lluís 40
Goenaga, I. 234
Gómez, Juan C. 224
Gottifredi, Sebastian 284
Granitto, Pablo M. 184, 224, 413
Grinblat, Guillermo L. 413
Guerra, Paulo T. 153
Guilherme da Silva, Fabio A. 133
Gutiérrez Estrada, Citlalih 144

Hernández, Yasmín 473

Ibargüengoytia, Pablo H. 463
Izetta, Javier 413

Kultz, Rene 163
Künzle, Luis Allan 163
Kuri-Morales, Angel 174

Lanzarini, Laura 60
Larese, Mónica G. 224
León, Elizabeth 214
Lopes Cardoso, Henrique 264
López-Camacho, Eunice 204
López, Javier 60
Lozano, Alexis 174
Ludermir, Teresa B. 396

Mahecha-Nieto, Isabel 214
Marcellino, Fernando J.M. 244
Marcos, M. Julieta 356

Martinez-Pena, Jorge 30, 572
Meléndez, Augusto 493
Méndez-Polanco, José Alberto 522
Meza, Ivan V. 20, 346, 512
Mineu, Nicole L. 396
Moguillansky, Martín O. 100
Morales, Eduardo F. 483, 493, 552
Morales-Manzanares, Eduardo F. 522
Morimitsu, Henrique 80
Muñoz, José 406
Muñoz-Meléndez, Angélica 522
Murrieta-Cid, Rafael 532

Nalepa, Gislaine M. 443
Navarro-Barrientos, Jesús Emeterio 433

Oliveira, Eugénio 264
Oliveira, Patricia R. 80
Osório, Fernando S. 50, 542
Osorio, Maria A. 562
Ossowski, Sascha 274

Paraboni, Ivandré 316, 326
Pessin, Gustavo 50
Pineda, Luis A. 10, 20, 346, 512

Rangel-Valdez, Nelson 30, 572
Rascón, Caleb 10, 512
Reyes-García, Carlos A. 1
Rendón Lara, Eréndira 144
Rodovalho, Wildener M. 296
Roman, Norton Trevisan 336, 376, 386
Ross, Peter 204

Salinas, Lisset 20, 346
Sánchez L., Abraham 562
Santos, André G. 70
Scalabrin, Edson E. 443
Scarton, Carolina 306
Schwind, Camilla 120
Sichman, Jaime S. 244
Sierra, B. 234
Silva, Fabiano 163
Silva Fagundes, Moser 274
Simari, Guillermo R. 90, 284, 356
Siqueira, Sean W.M. 133
Soares, Carlos 194
Soares, Michele S. 70
Soto, Ricardo 582
Sucar, Luis Enrique 473, 493, 552

Tenorio-Gonzalez, Ana C. 483
Terashima-Marín, Hugo 204
Toriz P., Alfredo 562
Torres-Jimenez, Jose 30, 572
Trigo, Paulo 254
Triviño, Gracián 406
Tuesta, Esteban F. 80

Urbano, Joana 264

Van Horebeek, Johan 433
Vázquez, Roberto A. 423
Venegas, Esther 346, 512
Villaseñor-Pineda, Luis 483
Vinhal, Cássio Dener N. 296

Wassermann, Renata 100, 153
Wolf, Denis F. 50, 542

Zapata, Rene 562
Zatarain-Cabada, Ramón 1
Zatarain-Cabada, Rosalío 1
Zelaia, A. 234