



**Maastricht University**



# **BNAIC 2012**

**Maastricht, October 25–26, 2012**

**Department of Knowledge Engineering**

## **Proceedings of BNAIC 2012**

The 24<sup>th</sup> Benelux Conference on Artificial Intelligence

Editors: Jos W.H.M. Uiterwijk, Nico Roos, and Mark H.M. Winands

BNAIC 2012  
The 24<sup>th</sup> Benelux Conference on  
Artificial Intelligence

PROCEEDINGS OF THE 24<sup>TH</sup> BENELUX CONFERENCE ON  
ARTIFICIAL INTELLIGENCE  
25 - 26 October 2012, Maastricht, The Netherlands

**Jos W.H.M. Uiterwijk, Nico Roos, and Mark H.M. Winands**

*The 24<sup>th</sup> Benelux Conference on Artificial Intelligence (BNAIC) 2012*  
Proceedings of the 24<sup>th</sup> Benelux Conference on Artificial Intelligence  
Jos W.H.M. Uiterwijk, Nico Roos, and Mark H.M. Winands (eds.)  
25 - 26 October 2012, Maastricht, The Netherlands

ISSN 1568-7805

Cover:

Design: Design-Studio Océ Business Services  
Front: Vrijthof (photography: Paul Mellaart)

Printing and binding: Océ Business Services

## Preface

The 24th edition of the Benelux Conference on Artificial Intelligence was held in Maastricht, The Netherlands, on October 25-26, 2012. It was organized by the Department of Knowledge Engineering of Maastricht University, which celebrates this year the 20th anniversary of its education programmes, the bachelor programme Knowledge Engineering and the master programmes Artificial Intelligence and Operations Research. It is a nice coincidence that 2012 also is the Turing year, on the occasion of the 100th birthday of Alan Turing, the famous British mathematician and founder of what is now known as Artificial Intelligence. Nowadays, we really can say that Artificial Intelligence is a flourishing research area, penetrating all aspects of computer science and becoming increasingly visible in daily life. Just think of all social media and internet activities, using all kinds of smart and intelligent devices. We believe that Turing would have loved it!

As usual, the BNAIC was organized under auspices of the Benelux Association for Artificial Intelligence (BNVKI), and the Dutch Research School for Information and Knowledge Systems (SIKS). The BNAIC is one of the main activities of the BNVKI and as stated in the regulations of the BNVKI, the main goals of the BNAIC are twofold: 1) to bring together the AI researchers in the Netherlands, Belgium, and Luxembourg, as a place to meet and to present research activities; and 2) to present high-quality research results, possibly already published in international conferences or journals. The format of the BNAIC is therefore a mixture of a meeting place and a forum for good-quality research results. This forms a balance that has proven to be successful in the previous years, as is shown by the high number of participants each year.

Let us give you some statistics. The number of submissions this year was 83, of which 41 were category A papers (original work), 37 category B (short abstracts of work already published elsewhere), and 5 category C (demonstrations). After reviewing by program committee members, 33 A papers were accepted (acceptance rate 80%), all but one of the B papers were accepted, and all demonstrations.

The conference consisted of 18 regular sessions, a poster session, a demonstrations session, and two keynote speeches, by world-renowned AI researchers. The first was given by Dr. Chris Welty from the IBM T.J. Research Center. He gave an insight in the development of WATSON, the first computer being able to outperform humans on general Question-and-Answer (Q&A) games, as demonstrated by its win in the famous *Jeopardy!* quiz against the 2 world-best human contenders. The second keynote speech was by Dr. Georgios Yannakakis from IT University of Copenhagen, who lead the way through past, present, and future of game AI. He delved in the reasons why academic game AI research hardly became visible in commercial applications thus far and elaborated on new promising game AI developments.

We acknowledge the enormous support we have received from many persons, companies and organizations, either financially, or by sponsoring awards, or by helping us in the organization. We thank the two keynote speakers for their willingness to share their insights and research experiences with our community. Further we are very grateful for all the programme committee members, who volunteered in reviewing the submissions. And last but not least, we thank the authors and the participants for their willingness to disseminate and receive inspiring ideas on this wonderful research area, Artificial Intelligence!

Jos Uiterwijk  
Mark Winands  
Nico Roos

Maastricht, October 2012

## BNAIC Conferences

1988	Amsterdam	2001	Amsterdam
1989	Twente	2002	Leuven
1990	Kerkrade	2003	Nijmegen
1991	Amsterdam	2004	Groningen
1992	Delft	2005	Brussels
1993	Twente	2006	Namur
1995	Rotterdam	2007	Utrecht
1996	Utrecht	2008	Enschede
1997	Antwerpen	2009	Eindhoven
1998	Amsterdam	2010	Luxembourg
1999	Maastricht	2011	Ghent
2000	Kaatsheuvel	2012	Maastricht

## Committees BNAIC 2012

### Conference Team

Hendrik Baier	Aude van den Bussche	Pim Nijssen	Mandy Tak
Daan Bloembergen	Daniel Claes	Frederik Schadd	
Haitham Bou Ammar	Claudine Jeurissen	Maarten Schadd	

### Programme Committee

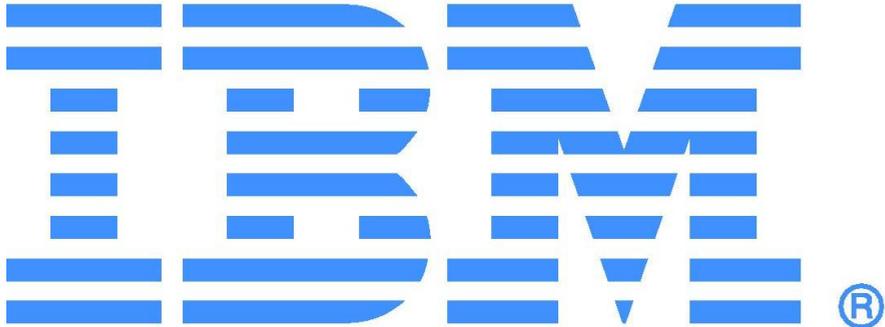
Ameen Abu-Hanna	Jeroen De Knijf	Tomas Klos	Johan Suykens
Huib Aldewereld	Mathijs De Weerd	Walter Kusters	Annette ten Teije
Geert Jan Bex	Virginia Dignum	Gerhard Lakemeyer	Dirk Thierens
Mauro Birattari	Frank Dignum	Peter Lucas	Leon van der Torre
Hendrik Blockeel	Hendrik Drachslers	Stijn Meganck	Paolo Turrini
Sander Bohte	Kurt Driessens	John-Jules Meyer	Karl Tuyls
Richard Booth	Linda van der Gaag	Ann Nowé	Jan Van den Bussche
Antal van den Bosch	Matteo Gagliolo	Wim Nuijten	Greet Vanden Berghe
Tibor Bosse	Benjamin Gateau	Mykola Pechenizkiy	Katja Verbeeck
Bruno Bouzy	Pierre Geurts	Eric Postma	Bart Verheij
Bert Bredeweg	Tim Grant	Rob Potharst	Sicco Verwer
Joost Breuker	Pascal Gribomont	Henry Prakken	Arnoud Visser
Joost Broekens	Marc Gyssens	Peter van der Putten	Peter Vranx
Maurice Bruynooghe	Frank van Harmelen	Jan Ramon	Natalie van der Wal
Martin Caminada	Frank Harmsen	Birna van Riemsdijk	Gerhard Weiss
Tristan Cazenave	Jaap van den Herik	Peter van Rosmalen	Emil Weydert
Tom Croonenborghs	Tom Heskes	Maarten Schadd	Marco Wiering
Walter Daelemans	Koen Hindriks	Marcin Seredynski	Floris Wiesman
Gregoire Danoy	Veronique Hoste	Evgueni Smirnov	Niek Wijngaards
Mehdi Dastani	Wojtek Jamroga	Ida Sprinkhuizen-Kuyper	Jef Wijsen
Martine De Cock	Uzay Kaymak	Thomas Stuetzle	Cees Witteveen

### Organizing Committee

Nico Roos	Department of Knowledge Engineering (DKE), Maastricht University
Mark Winands	Department of Knowledge Engineering (DKE), Maastricht University
Jos Uiterwijk	Department of Knowledge Engineering (DKE), Maastricht University

## Sponsors

**Main sponsor**



**Organized by**



**Maastricht University**

---

**Department of Knowledge Engineering**

## Other sponsors



[www.d-cis.nl](http://www.d-cis.nl)



[mssm.nl](http://mssm.nl)



*integrated ict*

[www.unilogic.nl](http://www.unilogic.nl)

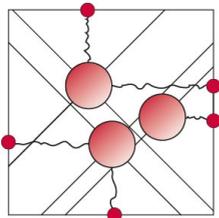


Netherlands Organisation for Scientific Research

[www.nwo.nl](http://www.nwo.nl)



[www.siks.nl](http://www.siks.nl)



*SNN Adaptive Intelligence*

[www.mlplatform.nl](http://www.mlplatform.nl)



Gemeente Maastricht

[www.maastricht.nl](http://www.maastricht.nl)



[www.maastrichtcongresbureau.nl](http://www.maastrichtcongresbureau.nl)

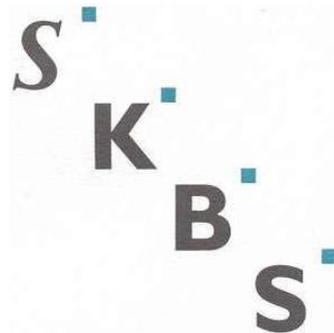


where to sleep what to do and how to travel

[www.maastrichtbookingservice.nl](http://www.maastrichtbookingservice.nl)



[www.unimaas.nl/bnvki](http://www.unimaas.nl/bnvki)



# Contents

<b>Preface</b>	<b>iii</b>
<b>Sponsors</b>	<b>v</b>
<b>Full Papers</b>	<b>1</b>
<b>Generating and Testing Multi-Issue Elections</b>	
Stéphane Airiau . . . . .	3
<b>Parallel Hybrid SAT Solving using OpenCL</b>	
Sander Beckers, Gorik De Samblanx, Floris De Smedt, Toon Goedeme, Lars Struyf, and Joost Vennekens . . . . .	11
<b>A Proof of the Expression of the Prior Convergence Error</b>	
Janneke Bolt . . . . .	19
<b>Multi-dimensional Classification with Naive Bayesian Network Classifiers</b>	
Janneke Bolt and Linda van der Gaag . . . . .	27
<b>Sharing Blood: A Decentralised Trust and Sharing Ecosystem based on the Vampire Bat</b>	
Wouter Bulten, Willem Haselager, and Ida Sprinkhuizen-Kuyper . . . . .	35
<b>On Extended Conflict-Freeness in Argumentation</b>	
Martin Caminada and Srdjan Vesic . . . . .	43
<b>A Nonparametric Evaluation of SysML-based Mechatronic Conceptual Design</b>	
Mohammad Chami, Haitham Bou Ammar, Holger Voos, Karl Tuyls, and Gerhard Weiss . . . . .	51
<b>Transfer Learning for Bilateral Multi-Issue Negotiation</b>	
Siqi Chen, Haitham Bou Ammar, Karl Tuyls, and Gerhard Weiss . . . . .	59
<b>Algorithms for Basic Compliance Problems</b>	
Silvano Colombo Tosatto, Marwane El Kharbili, Guido Governatori, Pierre Kelsen, Qin Ma, and Leon van der Torre . . . . .	67
<b>An Interactive Learning Approach to Histology Image Segmentation</b>	
Michael Derde, Laura Antanas, Luc De Raedt, and Fabian Guiza Grandas . . . . .	75
<b>Modeling and Evaluating Cooperation in Multi-Context Systems using Conviviality</b>	
Vasileios Efthymiou, Patrice Caire, and Antonis Bikakis . . . . .	83
<b>Automated Analysis of Social Choice Problems: Approval Elections with Small Fields of Candidates</b>	
Ulle Endriss . . . . .	91
<b>Greed, Envy, Jealousy - A Tool for more Efficient Resource Management</b>	
Mike Farjam, Willem Haselager, and Ida Sprinkhuizen-Kuyper . . . . .	99
<b>Preserving Precision as a Guideline for Interface Design for Mathematical Models</b>	
Linda van der Gaag, Hermi Schijf, Armin Elbers, and Willie Loeffen . . . . .	107
<b>Designing a Search and Rescue Simulation Environment for Studying the Performance of Agent Organizations</b>	
Mattijs Ghijsen, Wouter Jansweijer, and Bob Wielinga . . . . .	115
<b>A Generalization of the Winkler Extension and its Application for Ontology Mapping</b>	
Maurice Hermans and Frederik Schadd . . . . .	123
<b>Structure Approximation of Most Probable Explanations in Bayesian Networks</b>	
Johan Kwisthout . . . . .	131
<b>Topical Influence on Twitter: A Feature Construction Approach</b>	
Menno Luiten, Walter Kosters, and Frank Takes . . . . .	139
<b>Leveraging Unscheduled Event Prediction through Mining Scheduled Event Tweets</b>	
Florian Kunneman and Antal van den Bosch . . . . .	147
<b>Authorship Disambiguation and Alias Resolution in Email Data</b>	
Freek Maes and Johannes Scholtes . . . . .	155
<b>Depth-based Detection using Haar-like Features</b>	
Ruud Mattheij, Eric Postma, Yannick van den Hurk, and Pieter Spronck . . . . .	162
<b>Towards a Universal Change Detection Framework in Levees</b>	
Gabriel Mititelu, Robert-Jan Sips, Bram Havers, and Zoltán Szilávik . . . . .	170
<b>Reinforcement Learning for Energy-Reducing Start-Up Schemes</b>	
Kristof Van Moffaert, Yann-Michaël De Hauwere, Peter Vranx, and Ann Nowé . . . . .	178
<b>Comparing Context-Aware Routing and Local Intersection Management</b>	
Adriaan ter Mors and Cees Witteveen . . . . .	186
<b>Evaluation and Improvement of Laruelle-Widgrén Inverse Banzhaf Approximation</b>	
Frits de Nijis, Daan Wilmer, and Tomas Klos . . . . .	194

<b>A Tableau-based Reasoning Method for Four-valued Description Logic ALC</b>	
Wenzhao Qiao and Nico Roos . . . . .	202
<b>An Adaptive Stigmergic Coverage Approach for Robot Teams</b>	
Bijan Ranjbar-Sahraei, Gerhard Weiss, and Ali Nakisae . . . . .	210
<b>Dynamic Mechanism Design for Efficient Planning under Uncertainty</b>	
Joris Scharpff, Matthijs Spaan, and Mathijs de Weerd . . . . .	218
<b>Multiple Instance Learning using Bag Distribution Parameters</b>	
David M.J. Tax . . . . .	226
<b>A Norm Language for Distributed Organizations</b>	
Bas Testerink and Mehdi Dastani . . . . .	234
<b>Intercausal Cancellation in Bayesian Networks</b>	
Steven Woudenbergh, Linda van der Gaag, and Carin Rademaker . . . . .	242
<b>Shortest Path Gaussian Kernels for State Action Graphs: An Empirical Study</b>	
Saba Yahyaa and Bernard Manderick . . . . .	250
<b>Corpus-based Validation of a Dialogue Model for Social Support</b>	
Janneke van der Zwaan, Virginia Dignum, and Catholijn Jonker . . . . .	258

## Extended Abstracts

267

<b>Mapping Product Taxonomies in E-commerce</b>	
Steven Aanen, Lennart Nederstigt, Damir Vandić, and Flavius Frasinca . . . . .	269
<b>Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies</b>	
Tim Baarslag, Koen Hindriks, Mark Hendrikx, Alex Dirkzwager, and Catholijn Jonker . . . . .	271
<b>Nested Monte-Carlo Tree Search for Online Planning in Large MDPs</b>	
Hendrik Baier and Mark Winands . . . . .	273
<b>Providing Feedback for Common Problems in Learning by Conceptual Modeling using Expectation-Driven Consistency Maintenance</b>	
Wouter Beek and Bert Bredeweg . . . . .	275
<b>Credibility-Limited Revision Operators in Propositional Logic</b>	
Richard Booth, Eduardo Fermé, Sébastien Konieczny, and Ramon Pino Pérez . . . . .	277
<b>Linkage Neighbors, Optimal Mixing and Forced Improvements in Genetic Algorithms</b>	
Peter Bosman and Dirk Thierens . . . . .	279
<b>Modelling Collective Decision Making in Groups and Crowds</b>	
Tibor Bosse, Mark Hoogendoorn, Michel Klein, Jan Treur, Natalie van der Wal, and Arlette van Wissen . . . . .	281
<b>Reinforcement Learning Transfer via Sparse Coding</b>	
Haitham Bou Ammar, Karl Tuyls, Matthew E. Taylor, Kurt Driessens, and Gerhard Weiss . . . . .	283
<b>Recognizing Demand Patterns from Smart Card Data for Agent-based Micro-Simulation of Public Transport</b>	
Paul Bouman, Milan Lovric, Ting Li, Evelien van der Hurk, Leo Kroon, and Peter Vervest . . . . .	285
<b>Solving Satisfiability in Fuzzy Logics with Evolution Strategies</b>	
Tim Brys, Yann-Michaël De Hauwere, Martine De Cock, and Ann Nowé . . . . .	287
<b>The Log-Gabor Method: Speech Classification using Spectrogram Image Analysis</b>	
Harm Buisman and Eric Postma . . . . .	289
<b>Evolutionary and Swarm Computing for Scaling up the Semantic Web</b>	
Christophe Guéret, Stefan Schlobach, Kathrin Dentler, and Martijn Schut . . . . .	291
<b>Text-Based Information Extraction Using Lexico-Semantic Patterns</b>	
Frederik Hogenboom, Wouter IJntema, and Flavius Frasinca . . . . .	293
<b>Doubtful Deviations and Farsighted Play</b>	
Wojtek Jamroga and Matthijs Melissen . . . . .	295
<b>Norm Contextualization</b>	
Jie Jiang, Huib Aldewereld, Virginia Dignum, and Yao-Hua Tan . . . . .	297
<b>Automated Measurement of Spontaneous Surprise</b>	
Bart Joosten, Eric Postma, Emiel Kraemer, Marc Swerts, and Jeeseun Kim . . . . .	299
<b>Why Won't You Do What's Good for You? Using Intelligent Support for Behavior Change</b>	
Michel Klein, Nataliya Mogles, and Arlette van Wissen . . . . .	301
<b>Moment Constrained Semi-Supervised LDA</b>	
Marco Loog . . . . .	303
<b>Elitist Archiving for Multi-Objective Evolutionary Algorithms: To Adapt or Not To Adapt</b>	
Hoang Ngoc Luong and Peter Bosman . . . . .	305
<b>AU Classification using AAMs and CRFs</b>	
Laurens van der Maaten and Emile Hendriks . . . . .	307

<b>Playout Search for Monte-Carlo Tree Search in Multi-Player Games</b>	
Pim Nijssen and Mark Winands . . . . .	309
<b>Influence-Based Abstraction for Multiagent Systems</b>	
Frans Oliehoek, Stefan Witwicki, and Leslie Kaelbling . . . . .	311
<b>Enhancements for Monte-Carlo Tree Search in Ms Pac-Man</b>	
Tom Pepels and Mark Winands . . . . .	313
<b>Toward Machines that Behave Ethically Better than Humans Do</b>	
Matthijs Pontier and Johan Hoorn . . . . .	315
<b><i>k</i>-Optimal: A Novel Approximate Inference Algorithm for ProbLog</b>	
Joris Renkens, Guy Van Den Broeck, and Siegfried Nijssen . . . . .	317
<b>Tree-Based Solution Methods for Multiagent POMDPs with Delayed Communication</b>	
Matthijs Spaan and Frans Oliehoek . . . . .	319
<b>Lifted Variable Elimination with Arbitrary Constraints</b>	
Nima Taghipour, Daan Fierens, Jesse Davis, and Hendrik Blockeel . . . . .	321
<b>N-Grams and the Last-Good-Reply Policy applied in General Game Playing</b>	
Mandy Tak, Mark Winands, and Yngvi Björnsson . . . . .	323
<b>WebPIE: A Web-scale Parallel Inference Engine using MapReduce</b>	
Jacopo Urbani, Spyros Kotoulas, Jason Maassen, Frank van Harmelen, and Henri Bal . . . . .	325
<b>A Framework for Qualitative Multi-Criteria Preferences</b>	
Wietse Visser, Reyhan Aydoğan, Koen Hindriks, and Catholijn Jonker . . . . .	327
<b>Ordered Epistemic Logic: Semantics, Complexity and Applications</b>	
Hanne Vlaeminck, Joost Vennekens, Maurice Bruynooghe, and Marc Denecker . . . . .	329
<b>An Efficiently Computable Support Measure for Frequent Subgraph Pattern Mining</b>	
Yuyi Wang and Jan Ramon . . . . .	331
<b>Enhancing Predictability of Schedules by Task Grouping</b>	
Michel Wilson, Cees Witteveen, and Bob Huisman . . . . .	333
<b>Mechanism for Robust Procurements</b>	
Yingqian Zhang and Sicco Verwer . . . . .	335
<b>Agent Programming Languages Requirements for Programming Cognitive Robots</b>	
Pouyan Ziafati, Mehdi Dastani, John-Jules Meyer, and Leon van der Torre . . . . .	337
<b>A BDI Dialogue Agent for Social Support: Specification and Evaluation Method</b>	
Janneke van der Zwaan, Virginia Dignum, and Catholijn Jonker . . . . .	339

**Demonstrations** **341**

<b>User-Computer Persuasion Dialogue for Grounded Semantics</b>	
Martin Caminada and Mikołaj Podlaszewski . . . . .	343
<b>COCALU: Convex Outline Collision Avoidance under Localization Uncertainty</b>	
Daniel Claes, Daniel Hennes, and Karl Tuyls . . . . .	345
<b>Kamala in the Cloud: Bottom-Up Knowledge Modeling using the Topic Maps Data Model</b>	
Fiemke Griffioen, Gabriel Hopmans, Peter-Paul Kruijssen, and Quintin Siebers . . . . .	347
<b>Demonstration of eMate - Stimulating Behavior Change via Mobile Phone</b>	
Michel Klein, Nataliya Mogles, and Arlette van Wissen . . . . .	349
<b>Scheduling with Precedence Constraint Posting</b>	
Michel Wilson, Cees Witteveen, and Bob Huisman . . . . .	351

**List of Authors** **353**



**Full Papers**

**BNAIC 2012**



# Generating and testing multi-issue elections

Stéphane Airiau

*ILLC – University of Amsterdam, P.O. Box 94242 1090 GE Amsterdam*

## Abstract

Some collective decision problems involve deciding about multiple issues. Due to the combinatorial aspect of the problem, it may not be feasible to use a single election for deciding all the issues at once. Many approaches have been proposed, one of which proposes to organize a sequence of elections about few issues at a time. This solution is appealing as in each election, the voters have to provide their preferences on a small number of alternatives. However, if the size of the election is kept small, the preferential dependencies of some voters may be violated and a voter may not know how to vote. When some agents cannot vote, we want to know how to choose the sequence of elections so that the elected winner is a good candidate. We ran computer simulations using artificial data and we show that agendas that minimize the number of violations often elect a legitimate winner, which indicates that this method could be used in practice.

## 1 Introduction

Some collective decision problems consist of deciding about multiple issues. When all the issues are independent, the problem is simple. However, the preference for one issue may depend on the outcome of other issues. An illustration of such a dependence is the association between wine and food: someone prefers to have red wine over white wine when she is eating meat, but she prefers white wine over red when the main dish is fish. In the presence of dependencies, one solution is to use a single election in which the candidates are the combinations of the issues. This election has an exponential number of candidates, which is a problem for large number of issues. Compact representations for such preferences have been proposed (e.g., CP-networks [2]). There have been many important theoretical contributions on voting in combinatorial domains (e.g. pointing out paradoxes or proposing possible voting protocols [3, 9, 11, 4]).

One of the proposed solutions [7, 12, 1] is to partition the set of issues into smaller sets and organise a sequence of elections in which each set of issues is decided in turn. It is desirable to have small sets of issues: a voter reasons about a small set of candidates at a time (low computational cost) and the size of the ballot is not large (low communication cost). Voting issue by issue is possible in some cases [7], but sometimes it is impossible to decompose the election. The problem is to respect the dependencies of all voters at the same time. When the cost of computing preferences is of primary concern, we *do* want to use elections of small sizes at the expense of violating the preferential dependencies of some voters. Airiau et al. [1] have developed a theoretical framework to study this problem.

First, we argue that voting issue by issue is not often possible. If we extrapolate, this result suggests that in practice, having a small set of issues for each election is not likely to occur. Hence, if we want to impose sequential elections of small sizes for communication and complexity purposes, we will be forced to violate the preferential dependencies of some voters, i.e. some voters will not be able to voice their preferences. Using computer simulations, we want to estimate whether the winner of the election is sensitive, i.e., if a small number of voters cannot participate, is it the case that a “good” candidate is elected. The contribution of the paper is to show the winner of the election is likely to be legitimate. We first discuss possible voting strategies when the preference dependencies are not respected. Then we experimentally show that a careful choice of the agenda allows to often elect a legitimate winner.

## 2 Multi-Issue Election Framework

Let  $\mathcal{I} = \{1, \dots, m\}$  be a set of  $m \in \mathbb{N}$  issues. For each issue  $i \in \mathcal{I}$ , there is a finite set of alternatives  $D_i$ . An outcome of the election is a member of the combinatorial domain  $\mathcal{D} = D_1 \times D_2 \times \dots \times D_m$ . For a set of issues  $C \subseteq \mathcal{I}$ , we will denote  $\mathcal{D}[C] = \prod_{i \in C} D_i$ .

Let  $\mathcal{N} = \{1, \dots, n\}$  be a set of  $n \in \mathbb{N}$  voters. Each voter has a preference over  $\mathcal{D}$  and we will assume it is a *preorder*. This means that for any two outcomes, a voter may strictly prefer one over the other, she may be indifferent between them, or she may be unable to state a preference between them. The preference of a voter may have some structure, in particular, the preference may contain dependencies between issues, which is coined preferential dependencies. Informally, if issue  $i$  depends on issue  $j$ , a voter needs to know the outcome of issue  $j$  to state her preference about issue  $i$ . Let us consider a classical example: the choice of a menu at a restaurant. The issues are the choices of starter, main dish and wine. The domain is the set of all possible menus. A preferential dependence may be that the choice of wine depends on the choice of main course: when the main course is red meat, a client may prefer to have red wine over white wine. However, when the main course is fish, the same client will prefer a white wine. The formal definition follows:

**Definition 1** Issue  $i \in \mathcal{I}$  is **preferentially dependent** on issue  $j \in \mathcal{I}$  given preference relation  $\succeq$ , if there exist values  $x, x' \in D_i, y, y' \in D_j$ , and a vector of values  $\vec{z} \in \mathcal{D}[\mathcal{I} \setminus \{i, j\}]$  for the remaining domains such that  $x.y.\vec{z} \succeq x'.y.\vec{z}$  but  $x.y'.\vec{z} \not\succeq x'.y'.\vec{z}$ .

Preferential dependence induces an irreflexive directed graph on  $\mathcal{I}$ , with an edge from  $i$  to  $j$  whenever  $j$  depends on  $i$ . We will call this graph the *dependency graph*.

To elect an alternative from  $\mathcal{D}$ , the voters can choose one of the many voting procedures [10]. However, if the input of the voting rule involves a partial ranking over all possible candidates, such an approach quickly becomes infeasible as the number of candidates in  $\mathcal{D}$  is too large. One approach proposed in [1, 6] is to partition the set of issues and a linear order and run a sequence of elections. We call an **agenda** for the issues in  $\mathcal{I}$  to be a linear order on a partition of  $\mathcal{I}$ . In the following,  $\triangleright$  denotes an agenda. If  $I \subseteq \mathcal{I}$  is one of the subsets in the partition, then there will be a *local election* to choose an alternative from the domain  $\mathcal{D}[I]$ ; that is, all the issues in  $I$  will be voted on at the same time. The linear order  $\triangleright$  determines the order in which these local elections are held. The result of each local election is announced to all voters before the next local election takes place.

To summarize, the voting procedure unfolds as follows. First, each voter submits her dependency graph. Using an agenda selection function, one agenda  $\triangleright$  is chosen. Finally, local elections are run using a voting rule (e.g. the Borda rule or the plurality rule) using the order specified by the agenda. We will assume that the voting rules are resolute (i.e., they include a tie-breaking rule). After a local election is held, the results are announced to all voters before proceeding to the next local election.

When a voter is able to know her preferences for each election of an agenda, we will say that the agenda is *compatible* with her dependencies. For example, let us assume that an issue  $i \in \mathcal{I}$  depends on an issue  $j \in \mathcal{I}$  and the agenda is  $I_1 \triangleright I_2$ . When  $j \in I_1$  and  $i \in I_2$ , our voter is able to express her preferences as she will know the outcome of  $j$  before considering her preference about issue  $i$ . Also, if  $\{i, j\} \subseteq I_1$  or  $\{i, j\} \subseteq I_2$ , our voter will also be able to express her preferences (this time over combinations of issues including  $i$  and  $j$ ). In both cases, we say that the agenda is *compatible* with the preferential dependence of issue  $j$  on issue  $i$ . However, when  $i \in I_1$  and  $j \in I_2$ , then the voter cannot state her preference and we will say that the agenda *violates* the preferential dependency.

**Definition 2** An agenda  $I_1 \triangleright I_2 \triangleright \dots \triangleright I_p$  is **compatible** with the preferential dependency of  $j \in I_k$  on  $i \in I_l$  ( $k, l \in \{1, \dots, p\}$ ) when  $k \leq l$ . When the agenda is compatible with all preferential dependencies of a dependence graph, we say that the agenda is **compatible** with the dependency graph.

Let us denote  $G_i$  voter  $i$ 's dependency graph. The union  $\bigcup_{i \in \mathcal{N}} G_i$  gathers the dependencies of all the voters and we call it the *canonical graph*. An agenda that is compatible with the canonical graph is compatible with the dependencies of all voters. Using a condensation of  $\bigcup_{i \in \mathcal{N}} G_i$ , we obtain a partial order over a partition of  $\mathcal{I}$ . We call an agenda **canonical** when it is consistent with this partial order. A canonical agenda has two main properties: (1) it does not violate any preferential dependency, and (2) it minimizes the size of the largest election amongst all agendas without dependency violations.

Canonical agendas are not difficult to compute and we can easily design an algorithm that will run in quadratic time. If the size of the largest election is reasonably small, then this represents a good method of choosing voting agendas. Firstly because voters should not have large computational costs to know their preferences. Secondly, because the communication costs will be kept low: each voter will have to send some ballots containing a small number of candidates. A special case is when a canonical agenda contains only elections of size one (i.e., the agenda corresponds to voting issue by issue). Such agendas are called o-legal and elections with o-legal agendas are studied by Lang and Xia [7].

### 3 Some Lessons from Strict Linear Orders

We start by studying the agenda selection problem when preferences are strict linear orders. This is a standard assumption in social choice theory. We want to provide some intuitions about how often voting

issue by issue can be used. We first take a problem involving three issues  $A$ ,  $B$ , and  $C$ , i.e., there are  $2^3 = 8$  alternatives and  $8! = 40,320$  possible strict total orders. The number of potential dependency graphs is  $2^{3 \times 2} = 64$  (for each pair of issues  $(i, j)$ ,  $i \neq j$ ,  $i$  does or does not depend on  $j$ ). We wrote a program generating all the strict linear orders and their associated dependency graphs. We found out that two dependency graphs (the two simple cycles) do not correspond to the dependencies of any strict linear order. For them, one can check that any relation that satisfies these dependencies is cyclic, consequently, it cannot be a preference relation. In conclusion, when voters have strict preferences over three issues, there are 62 possible dependency graphs. We now describe them.

We identified all the patterns of the dependency graphs and counted how many strict linear orders correspond to each pattern. The results are summarised in Table 1. Furthermore, we indicate the number of instantiations of each pattern as well as the number of strict orders that correspond to each instantiation. For example, for the second pattern with two edges, there are 3 possible instantiations:  $A$  and  $B$  depend on  $C$ ,  $A$  and  $C$  depend on  $B$ , and  $B$  and  $C$  depend on  $A$ . For each of these instantiations, we counted 32 different strict linear orders that satisfy exactly these dependencies.

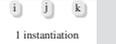
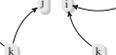
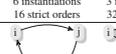
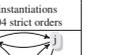
0 edges		1 instantiation 384 strict orders	1 edge		6 instantiations, 672 strict orders			
2 edges		6 instantiations 16 strict orders		3 instantiations 32 strict orders		3 instantiations 512 strict orders		3 instantiations 608 strict orders
3 edges		2 instantiations no strict orders		6 instantiations 120 strict orders		6 instantiations 216 strict orders		6 instantiations 384 strict orders
4 edges		6 instantiations 48 strict orders		3 instantiations 656 strict orders		3 instantiations 1200 strict orders		3 instantiations 1504 strict orders
5 edges		6 instantiations 6,912 strict orders	6 edges			1 instantiation 14,112 strict orders		

Table 1: Distribution of the strict orders over all the dependency graph patterns.

The table first suggests that there is a small proportion of strict linear orders that have an acyclic dependency graph (6,864 preferences, i.e. 17.02% of all strict linear orders)<sup>1</sup>. To vote issue by issue, each voter must have one of these preferences, and then dependency graphs of all voters must agree. Now, let us take an agenda  $\triangleright$  corresponding to voting issue by issue. Using the table, we found that there are 3080 different strict linear orders that are compatible with this agenda<sup>2</sup>, i.e. 7.64% of all possible strict linear orders. This means that if there are 10 voters and all preferences are equally likely (impartial culture assumption), the probability that the voters can all express their preferences is extremely tiny ( $6e^{-12}$ ). This is probably not so surprising as the assumption is quite strong, nevertheless, the probability of having an o-legal profile seems quite small for three issues.

We now consider the case of more than three issues. We assume that all the strict orders are equally likely. When the number of issues becomes larger, it gets more likely that a given issue depends on another issue. In Table 2, we provide the likelihood that the dependency graph of a given strict preference order is the full graph (i.e., if one picks randomly a strict linear order, we provide the likelihood that the corresponding dependency graph is the full graph). For 2 and 3 issues, we used an enumeration of all linear orders and for 4 and 5 issues, we estimated the likelihood by sampling 40,000 strict linear orders and by computing their corresponding dependence graph. With 5 issues, the likelihood that a voter has a full dependency graph is already over 90%. This suggests that voting issue by issue will not be possible under these conditions (although of course, the impartial culture assumption is quite restrictive).

# of issues	2	3	4	5
proportion of s.o. with full graph	$\frac{1}{3}$	$\frac{7}{20}$	0.578	0.9345

Table 2: All strict orders are equally likely: as the number of issues gets larger, the dependency graphs gets more likely to be the full graph.

<sup>1</sup>In Table 1, there are 6 acyclic patterns: the one with no edges (384 preferences), the one with a single edge (6 instantiations, each corresponding to 672 preferences, a total of  $6 * 672$  preferences), three patterns with two edges ( $6 * 16 + 3 * 32 + 3 * 512$  preferences), and one pattern with 4 edges ( $6 * 120$  preferences).

<sup>2</sup>For each pattern, we enumerated all instantiations and checked which ones are compatible with  $\triangleright$ . For the pattern with three edges, there is a single compatible instantiation (if  $\triangleright$  is  $\{i\} \triangleright \{k\} \triangleright \{j\}$ , then the instantiation in Table 1 is the one). Hence, we count 120 preferences for this pattern.

## 4 Some Difficulties with Using Preorders

Agents may lack the cognitive and computational resources to fully rank all alternatives. Hence, it may be more appropriate to represent preferences as preorders. We now point out some difficulties about using preorders in practice. In Figure 1, we provide an example of two preferences over three issues  $A$ ,  $B$  and  $C$  represented in two ways: as a CP-network (a compact representation of preorders), and as naive representation (an edge  $(x, y)$  means that  $x \succ y$ ). A voter with the left preference prefers the alternative  $abc$  over any other alternative and she cannot compare some alternatives, for example  $abc$  with  $a\bar{b}\bar{c}$ .

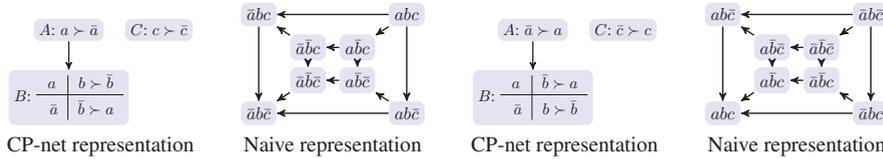


Figure 1: Two preferences over three binary issues  $A$ ,  $B$  and  $C$ .

We need to adapt the definition of voting rules to preorders [5]. The Borda rule is defined for strict orders (each candidate gets a unique score). Since there can be indifferences or incomparabilities, we cannot assign a unique score to each candidate. We define the score of a candidate as the number of candidates she dominates. In the example of Figure 1,  $a\bar{b}\bar{c}$  obtains a score of 1 for the left preference as  $a\bar{b}\bar{c}$  dominates only one other candidate, whereas  $a\bar{b}\bar{c}$  obtains a score of 5 in the preference on the right.

Surprisingly, we notice that two agendas compatible with the dependencies of all the voters can elect different winners! The previous example in Figure 1 illustrates this phenomenon. The two voters share the same dependency graph, but they have opposite preferences. Let us first consider the case of voting issue by issue, for example  $\{A\} \triangleright \{B\} \triangleright \{C\}$ , which is compatible with the dependency graph of both voters. We have a tie for each election, and let us assume that the tie-breaking rule chooses  $\bar{a}$  over  $a$ ,  $\bar{b}$  over  $b$  and  $\bar{c}$  over  $c$ . The winner ends up being  $a\bar{b}\bar{c}$ . Now, let us consider the case of voting on the three issues at once and use the Borda rule. Using the naive representation, we see we have a tie between  $abc$  and  $a\bar{b}\bar{c}$ , both obtaining a Borda score of 7. The issue-by-issue winner  $a\bar{b}\bar{c}$  obtains a Borda score of 6. This causes the problem of defining what should be a winner in such an election. In both agendas, all the voters could express their preferences so the two winners have some legitimacy. In the following, we will consider that a winner of any agenda that is compatible with the dependencies of all voters is legitimate.

**Definition 3 (legitimate winner)** For a given voting rule, a winner is **legitimate** if it is one winner of an agenda that is compatible with the dependencies of all voters.

## 5 Bounding the Size of the Largest Election

In this section, we allow agendas that violate the dependency graph of some voters, which is very likely to occur if we impose local elections of small sizes. The consequence is that some voters are uncertain about their preferences as one of the current issues depends on an issue that has not been settled yet. Let us denote  $\mathcal{G}_k(\mathcal{I})$  the set of agendas such that the largest election size is  $k$ . Let us take  $I_1 \triangleright I_2 \triangleright \dots \triangleright I_p$  an agenda in  $\mathcal{G}_k(\mathcal{I})$ . Let us assume that we are in the  $t^{\text{th}}$  election  $I_t$ . We note by  $F_t$  the set of issues that have already been fixed and by  $O_t$  the set of open issues (i.e., that will be decided later). More formally, we have  $F_t = \bigcup_{\tau=1}^{t-1} I_\tau$  and  $O_t = \bigcup_{\tau=t+1}^p I_\tau$ .

Suppose that a voter  $v \in N$  is uncertain. There is at least a pair of issues  $(j, i) \in G_v$  such that  $i \in I_t$  is in the current election and depends on  $j \in O_t$  that will be decided later. Let us call  $Q_t$  the set of issues that are missing to know her preferences for the current election, i.e.,  $Q_t = \{j \in O_t \mid \exists i \in I_t, (j, i) \in G_v\}$ .

As an example, suppose that the agenda is  $\{B, C\} \triangleright \{A\}$  and that we have two voters whose preferences are represented in Figure 1. This agenda is not compatible with the dependency graphs (but maybe it is with the ones of many other voters). For the preferences on the left, as  $c \succ \bar{c}$ , her choice is between  $bc$  and  $\bar{b}\bar{c}$ . In the case in which  $a$  is chosen, she prefers  $abc$  to  $a\bar{b}\bar{c}$ , however, if  $\bar{a}$  is chosen, then she prefers  $a\bar{b}\bar{c}$  to  $a\bar{b}\bar{c}$ . Note that this information is easily readable from the CP-net. A voter may decide to abstain from voting, or she can try to vote strategically. Among many possibilities, we will focus on the following two strategic behaviours. She can be “optimistic”, in which case, the voter hopes that  $a$  will win, and then she should vote for  $bc$ . She can also be “pessimistic” by expecting the worst case scenario, in this case  $\bar{a}$  wins, and then she should vote for  $\bar{b}\bar{c}$ . In the following, we will start by considering that the voters have preferences that are represented by acyclic CP-networks. Then, we will continue with the more general case of CP-networks.

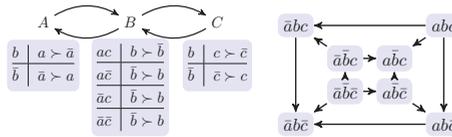


Figure 2: Example of a cyclic CP-network.

## 5.1 When Preferences are Acyclic CP-networks

We assume that the preference of a voter  $v$  is represented using an acyclic CP-net. Since the dependency graph is acyclic, we can take one topological ordering  $\triangleright_v$  of the set of issues. We can view  $\triangleright_v$  as an agenda in which the voter  $v$  can vote issue by issue (it may be different from the agenda of the sequential election, but  $\triangleright_v$  will come in handy for reasoning). Suppose that a voter is uncertain for election  $I_t$ . An uncertain voter can take one of the following three behaviours:

**Abstention** Since the voter does not have enough information to make a choice, one solution is for her to abstain. The voter may still participate in a later election.

**Optimistic Voter** One uncertain voter may be “wishful thinking” and hope that the outcomes of future elections are in her favour (in particular the elections include issues in  $Q_t$ ). In this case, the voter can make her guess in polynomial time. The idea to guess the missing issues in  $Q_t$  is to perform a *forward sweep* as is done for query optimisation [2]. Using  $X_1 \triangleright_v X_2 \cdots \triangleright_v X_m$  ( $X_i \in \mathcal{I}$ ), we go over each issue in turn until all issues in  $Q_t$  are decided. For a given issue  $X_i$ , either it has already been decided in a previous election, in which case we set  $X_i$  to the corresponding value. Or the value has not been decided, in which case we instantiate it with its maximum value given the instantiation of its parents (and using  $\triangleright_v$ , we are guaranteed they have been set). Once all the issues in  $Q_t$  are decided, a voter can express her preferences for  $I_t$  (e.g., by computing the Borda score).

**Pessimistic Voter** The goal of the pessimistic voter is to vote for an alternative that will improve the worst possible outcome. We can use the algorithm for optimistic using an opposite preference.

## 5.2 When Preferences are General CP-networks

We now allow cyclic dependence graphs. When there is a cycle between a set of issues  $C$ , a voter cannot express her preference for one issue in  $C$  without knowing the outcome of the other issues in  $C$ . For an agenda in  $\mathcal{G}_k(\mathcal{I})$  (the size of any election is bounded by  $k$ ), a voter whose dependency graph has a cycle of size  $l > k$  will be uncertain at some point. Using sequential elections may not be a good mechanism with such voters. Figure 2 is an example of such a preference with three issues.

With a cyclic CP-network, there is a combinatorial explosion when one wants to make a guess. Suppose a voter has the preferences represented in Figure 2 and that the agenda starts with an election about issue  $B$  only. Using the full preferences, we see that the best element is  $\bar{a}\bar{b}\bar{c}$  and the worst are  $\bar{a}b\bar{c}$  and  $\bar{a}bc$ . An optimistic behaviour would then be to wish that  $\bar{a}$  and  $\bar{c}$  get elected and then vote in favour of  $\bar{b}$ . For the pessimistic behaviour, the voter is indifferent between  $b$  and  $\bar{b}$ . The example shows that we can define a strategic behaviour, however, we needed to build the *full* preferences of the voter instead of reading the CP-table, which is computationally expensive. Since one of our goals is to ensure small computational efforts for the voter, this solution is not appropriate. Therefore, in case of general CP-networks, the only attitude we shall consider is abstention.

## 6 Experiments

We consider that a voter has the computational capacity to compute the winners of small elections. In our experiments, we assume that a voter can vote in elections about three binary issues at most (i.e., she can rank up to 8 candidates). When the canonical agenda has small size elections (i.e., when it is in  $\mathcal{G}_3(\mathcal{I})$ ), a legitimate winner will be elected, hence we can use this agenda. Otherwise, it is not clear which agenda to choose, and the number of agendas available is exponential in the number of issues. We would like to select an agenda that is likely to elect a legitimate winner. Intuitively, an agenda that minimizes the number of violations of dependencies should give a “good” winner. In the following experiments, we want to confirm our intuition and derive a good policy for selecting an agenda.

## 6.1 Assumptions and Data Generation

Unfortunately, we do not have access to any real data and generating realistic preferences is a difficult problem. The CATS system [8] generates preferences for combinatorial auctions, but there is no such systems for voting in multi-issue domains. We cannot use CATS preferences as they are monotone (e.g., having one more item is always preferred), which is not the case in our domain (e.g. a voter may prefer that one project is accepted, but not two). We propose one method for generating preferences, and of course other methods are also acceptable. As a starting point, we argue that the method is reasonable.

We first assume that there is some common pattern in the dependency graph. For some multi-issue elections some “true”, objective dependencies may exist. Some voters may be well-aware of them, when others (due to lack of knowledge or misinterpretations) may make mistakes. Another interpretation of that assumption is that many voters agree on some dependencies while a few other dependencies are not shared by all voters. To model this assumption, the dependency graphs of all the voters are perturbations of one dependency graph  $G_o$ . To generate  $G_o$ , we first pick a random number of edges from a uniform distribution, and then we place the edges randomly in the graph (for experiments with acyclic graphs, we do so as long as their addition does not form a cycle). Then, we generate the other dependency graphs by perturbing  $G_o$ . For each potential edge, if it is an edge in  $G_o$ , we remove it with probability  $r_1$ , otherwise, we add it with probability  $r_2$  (only if the addition of the edge does not create a cycle for experiments with acyclic graphs, which may produce a slight bias towards removing edges from  $G_o$ ). For the experiments, we use  $r_1 = r_2 = 0.2$ . We generate random acyclic preferences compatible with each dependency graph (i.e., we ensure that the preferences contain all the dependencies present in the dependency graph).

In all the following experiments, we use the Borda rule for local elections. For a preference profile, we need to form the set of legitimate winners. To do so, we use our Borda rule without tie breaking to compute the winners of two agendas: the agenda consisting of a single election about all the issues and a canonical agenda. The union of these two sets forms a set of legitimate winners<sup>3</sup>

## 6.2 Results with Acyclic Dependency Graphs

We first ran experiments with  $|\mathcal{I}| = 5$  binary issues,  $|\mathcal{N}| = 10$  agents, and we average over 500 preference profiles. In 28% of the preference profiles generated, the largest election of the canonical agenda is less than 3, hence it produces a legitimate winner.

For the remaining profiles, we generated all agendas in  $\mathcal{G}_3(\mathcal{I})$  (with 5 issues, there are 530 agendas in  $\mathcal{G}_3(\mathcal{I})$ ). For each of these agendas, we compute the winner when all the voters use an optimistic, pessimistic behaviour, or abstain. Over all the agendas in  $\mathcal{G}_3(\mathcal{I})$ , there is on average about 15 different winners<sup>4</sup>, one of them being legitimate. With 5 binary issues, there are 32 candidates. This means that if an agenda is selected using a uniform distribution, about half of the candidates can be elected when some voters are uncertain! We also counted how many times a winner is legitimate. On average, a legitimate winner is elected in about 29% of the time<sup>5</sup>. Adding the preference profiles for which a canonical agenda is in  $\mathcal{G}_3(\mathcal{I})$ , a legitimate winner is elected 49% of the time.

For each agenda in  $\mathcal{G}_3(\mathcal{I})$ , we computed the number of violations (for each dependence graph of each voter, we count one violation when a voter needs one issue that will be set in a future election). There can be at most  $10 * 5 * 4 = 200$  violations in total (if all issues depend on all other issues). Figure 3(a) provides the number of agendas for a given number of violations. We see that on average, very few agendas have very few or very many violations and there is a peak at about 20 violations. We also indicate how many of these agendas elect a legitimate winner. From the same data, Figure 3(b) displays the proportion of agendas electing a legitimate winner. We observe that an agenda with low number of violations is more likely to elect a legitimate winner (although at best, in only about 55% of the cases a legitimate winner is elected). Abstaining from voting or the optimistic attitude yield similar outcomes, the pessimistic attitude performs slightly worse.

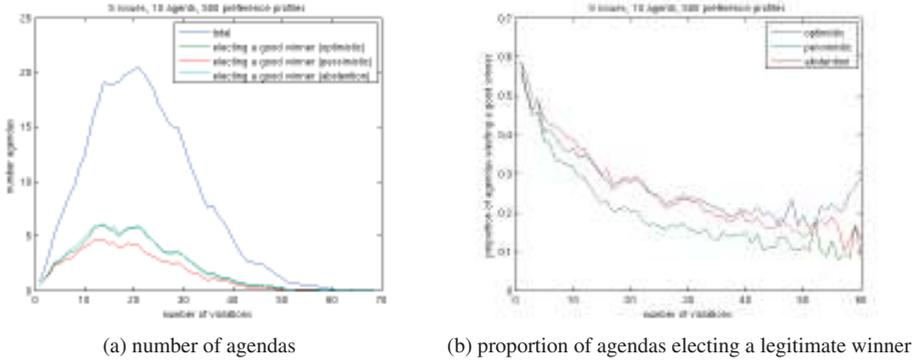
We would like to assess the quality of a non-legitimate winner. Figure 4 shows the Borda score (computed for an election with all the issues) of the winners<sup>6</sup>. The mean of the Borda score per voter for a legitimate winner is 17.1 (i.e., on average legitimate winners dominate 17.1 candidates (out of 31)). This means that the elections we generated are quite close. If the winner is not legitimate, she obtains a lesser Borda score (these winners dominate 2.8 candidates less). This difference in Borda score is not very large, so the winner, though she may not be completely legitimate, may be acceptable. Finally, one important observation is that abstaining or using optimistic behaviour yield similar results and using pessimistic

<sup>3</sup>it is a subset of all legitimate winners, due to computation costs, we did not compute the set, so the results in the paper are pessimistic.

<sup>4</sup>There are 14.7 winners for optimistic voters (std 4.7), 18.1 for pessimistic ones (std 4.3) and 13.7 when they abstain (std 4.0).

<sup>5</sup>A legitimate winner is elected in 29.8% with optimistic voters, 22.0% with pessimistic ones and 29.3% when they abstain.

<sup>6</sup>The curve for the legitimate winners fluctuates because the legitimate winners may have different Borda scores and they are not elected equally often.

Figure 3: Results over all agendas in  $\mathcal{G}_3(\mathcal{I})$ .

behaviour yields worse outcomes in general. With a pessimistic behaviour, we already noticed that there were more winners over all the agendas. We now notice that legitimate candidates are elected less often. Also, when a non-legitimate candidate is elected, the average Borda score is less than with optimistic behaviour or abstention. This fact shows that the uncertain voters have some power in the decision. But we do not have a clear argument to explain why pessimistic has a negative impact compared to abstaining when optimistic has no clear advantage.

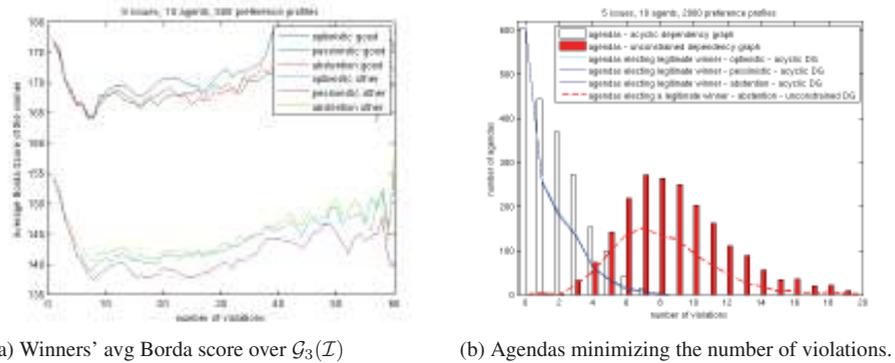


Figure 4: Borda score and agendas minimizing the number of violations

Agendas minimizing the number of violations elect more often legitimate winners. We focus on these agendas in the next experiments. We generated 2,000 preference profiles and for each preference profile we select the agendas that minimize the number of violations. Figure 4(b) provides the distribution of the minimum number of violations over all the preference profiles. As in the previous experiments, a canonical agenda can be used in about 30% of the preference profiles. Then, the figure shows that we always found at least one agenda with at most 9 violations for any profile. Over all profiles for which a canonical agenda is not in  $\mathcal{G}_3(\mathcal{I})$ , a legitimate winner is elected in about 50% of the times<sup>7</sup>. One possibility to explain this result is that with a low number of violations, many voters can express their preferences and the uncertain voters do not have much voting power. Finally, note that if we take into account the profiles when a canonical agenda is in  $\mathcal{G}_3(\mathcal{I})$ , a legitimate winner is elected 65% of the time.

In conclusion, selecting an agenda that minimizes the number of violations leads better results: it elects a legitimate winner in about 65% of the time and will elect an “acceptable candidate” the rest of the time. The agendas that minimise the total number of violations are the ones containing elections about three issues at once. In the future, we should be able to focus only on those agendas.

<sup>7</sup>A legitimate winner is elected in 49.8% of the times with optimistic voters, in 48.6% when they are pessimistic and in 50.6% when they abstain.

### 6.3 Results with General Dependency Graphs

We ran experiments with general CP-networks. A first difference is none of the canonical agendas is in  $\mathcal{G}_3(\mathcal{I})$ . Since some cycles may already exist in the dependence graphs of the voters, it is less likely that any canonical agenda contains elections of small sizes. We also observed that agendas had in general more violations (we do not include graphs such as the one of Figure 3 and 4 for lack of space). We found out that a legitimate winner was elected in 28.3% over all agendas in  $\mathcal{G}_3(\mathcal{I})$  and that agendas with small number of violations elect more often a legitimate winner. For the agendas that minimize the number of violations (see in Figure 4), we observe that the minimum number of violations are higher. Nevertheless, in 49.5% of the time, a legitimate winner is elected.

## 7 Conclusion and Future Work

We studied the problem of voting in multi-issue elections using computer simulations. We argued that an agenda containing elections of small sizes is likely to violate the preferential dependencies of some agents. We studied the agenda selection problem experimentally with 6 issues and imposing that each local election is about at most 3 binary issues (i.e. no more than 8 candidates). We provided arguments in favour of choosing an agenda minimizing the number of dependence violations. For future work, we would first like to identify other ways of generating preferences. Then, we would like to investigate decision problems involving a lot more issues. Considering all possible agendas is not feasible, but our results suggest we can concentrate on agendas that minimize the number of elections. Finally, we would like to use pattern recognition techniques to find a better criterion for selecting agendas.

## References

- [1] S. Airiau, U. Endriss, U. Grandi, D. Porello, and J. Uckelman. Aggregating dependency graphs into voting agendas in multi-issue elections. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*, July 2011.
- [2] C. Boutilier, R.I. Brafman, C. Domshlak, H.H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.
- [3] S.J. Brams, D.M. Kilgour, and W.S. Zwicker. The paradox of multiple elections. *Social Choice and Welfare*, 15(2):211–236, 1998.
- [4] V. Conitzer, J. Lang, and L. Xia. Hypercubewise preference aggregation in multi-issue domains. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 158–163, 2011.
- [5] U. Endriss, M.S. Pini, F. Rossi, and K.B. Venable. Preference aggregation over restricted ballot languages: Sincerity and strategy-proofness. In *Proc. 21st International Joint Conference on AI (IJCAI-2009)*, 2009.
- [6] D. Lacy and E.M.S. Niou. A problem with referendums. *Journal of Theoretical Politics*, 12(1):5–31, 2000.
- [7] J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57:304–324, 2009.
- [8] K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auction algorithms. In *the Proceedings of ACM Conference on Electronic Commerce*, 2000.
- [9] F. Rossi, K.B. Venable, and T. Walsh. mCP Nets: Representing and reasoning with preferences of multiple agents. In *Proc. 19th AAAI Conference on AI (AAAI-2004)*, 2004.
- [10] A.D. Taylor. *Social Choice and the Mathematics of Manipulation*. Cambridge University Press, 2005.
- [11] L. Xia, V. Conitzer, and J. Lang. Voting on multiattribute domains with cyclic preferential dependencies. In *Proc. 23rd AAAI Conference on AI (AAAI-2008)*, 2008.
- [12] L. Xia, V. Conitzer, and J. Lang. Strategic sequential voting in multi-issue domains and multiple-election paradoxes. In *Proceedings of the 12th ACM Conference on Electronic Commerce (EC-11)*, pages 179–188, 2011.

# Parallel hybrid SAT solving using OpenCL

Sander Beckers<sup>ab</sup>

Gorik De Samblanx<sup>a</sup>

Floris De Smedt<sup>a</sup>

Toon Goedemé<sup>a</sup>

Lars Struyf<sup>a</sup>

Joost Vennekens<sup>ab</sup>

<sup>a</sup> *EAVISE, Lessius - Campus De Nayer, Sint-Katelijne-Waver, Belgium*

<sup>b</sup> *CS-DTAI, KULeuven, Leuven, Belgium*

## Abstract

In the last few decades there have been substantial improvements in approaches for solving the Boolean satisfiability problem. Many of these consisted in elaborating on existing algorithms, both on the side of complete solvers as in the area of incomplete solvers. Besides the improvements to existing solving methods, however, recent evolutions in SAT solving take the form of combining several solvers into one, resulting in parallel solvers and so-called hybrid solvers. Our goal is to combine both approaches, by presenting a parallel hybrid solver. The parallelism exists on two levels: we run a complete solver on the CPU concurrently with an incomplete solver on the GPU, where the latter in turn consists of a massively parallel local search algorithm. We implemented our approach using the OpenCL framework, and present preliminary experimental results.

## 1 Introduction

The last few decades have seen an enormous amount of progress on solving the Boolean satisfiability problem. Much of the progress in this area was due to elaborations of existing algorithms. On the side of the complete solvers this led to more efficient branching heuristics, the use of watched literals for unit propagation and conflict driven clause learning; incomplete solvers on the other hand have benefited from using random walks and from integrating evolutionary algorithms into the search process. Besides the improvements to existing solving methods, however, many of the evolutions in SAT-solving over the last decade resulted from coming up with ways to combine several solvers into one. These new type of solvers are based on the more traditional ones, but add a novel element by making several of them cooperate (or compete, in the case of portfolio strategies as in [5, 9]).

We can, roughly speaking, separate these cooperative approaches into two classes. On the one hand we have hybrid solvers, the motivation for which comes from attempts to respond to one of the Ten Challenges in Propositional Reasoning and Search posed by Selman et al. [20], namely

CHALLENGE 7 : Demonstrate the succesful combination of stochastic search and systematic search techniques, by the creation of a new algorithm that outperforms the best previous examples of both approaches.

The idea is that if we succeed in combining ideas from complete and incomplete methods, the deficiencies of the one method are complemented by the advantages of the other in order to obtain the best of both worlds.

On the other hand, with gains in processor speeds becoming limited and the resulting advent of multi-core processors, a lot of work has been done on parallel solvers. Usually these take the form of divide-and-conquer strategies, where the search space is divided over several sequential solvers, who work together by sharing learned clauses, as is the case in [6, 10].<sup>1</sup>

Both approaches have already resulted in substantial improvements by themselves, however in our opinion they still fall short in two ways. First of all, none of the existing state-of-the-art parallel solvers make

---

<sup>1</sup>Notable exceptions are the winners of the 2009 and 2011 SAT competition parallel track, ManySAT and Plingeling, which use a portfolio approach [5, 9].

use of the possibilities that have been opened up over the last couple of years by General Purpose GPU-programming. Typical parallel solvers make use of four or eight threads working concurrently, whereas the massive parallelism offered by GPUs nowadays allows up to thousands of threads. Second of all, these lines of research have evolved separately from each other. Taking our cue from the seventh SAT-challenge and taking it a step further, one can conjecture that combining the algorithms from both approaches into one framework is likely to be mutually beneficial. This work forms the first effort in overcoming these shortcomings. It presents a solver that is both hybrid as well as massively parallel.<sup>2</sup>

Our strategy is the following: we start out with parallelizing the basic hybrid solving method presented in [14], where the trace of an incomplete solver is used to guide a complete solver. Rather than sequentially calling the complete solver after every iteration of the incomplete solver, we run both at the same time. Furthermore, we will run a massively parallel adapted version of the used local search algorithm. To be more concrete, we implemented these ideas using the OpenCL framework. Our solver consists in running MiniSAT on the CPU, while a random walk algorithm will run on the GPU from which we derive a heuristic to guide MiniSAT, replacing its normal VSIDS-like heuristic.

The next section will first provide a very basic overview on SAT solvers and then introduces the ideas behind hybrid solvers. We will focus on the work done by Mazure and his colleagues, the success of which serves as the justification for our methodology. We then proceed, in section 3, to outline our approach. First we motivate the CPU-GPU implementation, after which we introduce the OpenCL framework. In subsection 3.3 we sketch the details of the OpenCL-SAT solver. The results on a set of test-cases are discussed in section 4. In conclusion, we reflect on the possibilities our research presents for future work.

## 2 SAT solving essentials

### 2.1 Background on SAT solvers

Traditionally there have been two types of solvers for the SAT problem, namely complete solvers and incomplete solvers. Complete solvers are guaranteed to decide for every SAT instance whether or not it is satisfiable, given enough time, whereas incomplete solvers may stop before finding a solution even if one exists. The benefit of the latter is that they are in general a lot faster on satisfiable instances than complete solvers, but they have the obvious disadvantage that they are incomplete. To repair this defect and have the best of both worlds, hybrid solvers were developed. These combine both types of solvers in a way that tries to retain as much as possible of their respective advantages.

The backbone of all complete solvers that have been developed over the last few decades remains the DPLL algorithm that was discovered in the 60's [7]. The algorithm is a recursive, depth-first enumeration of all possible assignments in the model space, which can be seen as a binary tree. The progress in this area was mainly due to intelligently optimizing the different parts of the algorithm, such as the heuristics used to choose the next branch in the search tree, and the data structures used for the propagation of unit literals. An additional improvement came from expanding the original problem by adding new clauses that have been learned from conflicting assignments (so-called Conflict Driven Clause Learning, CDCL). For our research the branching heuristic will be most significant. It determines in which order the search tree is traversed, and choosing a relevant heuristic can therefore drastically reduce the number of recursive calls to be made in order to find a solution. A lot of modern SAT solvers use variants of the Variable State Independent Decaying Sum (VSIDS) heuristic [17], which assigns a score to each literal according to its activity in the search process.

In contrast to complete solvers, a typical incomplete solver (as described in [18]) uses a greedy local search algorithm, where the variables are initiated with a random assignment, and at each iteration a single variable is 'flipped' to its opposite value. The variable to be flipped is chosen such that the number of satisfied clauses is maximized. This continues until either a solution has been found, or a fixed limit of flips has been performed. In the latter case, the process is repeated, for a set number of tries. To avoid ending up in local minima, the greedy approach is usually combined with a random walk, so that with a certain probability a variable that appears in an unsatisfied clause is chosen at random instead of the greedy choice [19]. This simple algorithm performs well on a wide set of instances, but is not guaranteed to find a solution.

---

<sup>2</sup>At an earlier stage of our research, the outline of our approach appeared already in [4].

---

**Algorithm 1 DP+TSAT: basic version**

---

**Procedure**  $DP+TSAT$ **Input** : a set of clauses  $S$ **Output** : a satisfying truth assignment of  $S$ , if found  
or a definitive statement that  $S$  is inconsistent**Begin**Unit\_propagate( $S$ );**if** the empty clause is generated **then return** (false);**else if** all variables are assigned **then return** (true)**else begin****if** TSAT(  $S$  ) succeeds **then return** (true)**else begin** $p :=$  the most often falsified literal during TSAT;**return** ( $DP+TSAT(S \wedge p) \vee DP+TSAT(S \wedge \neg p)$ );**end;****end;****End**

---

## 2.2 The hybrid approach

Motivated by the seventh challenge mentioned in the introduction, Mazure et al. [1, 2, 3, 8, 11, 12, 13, 14, 15] have investigated several ways of combining complete and incomplete solvers. Mazure demarcates three types of hybrid approaches to be found in the literature [15].

1. DPLL-based solvers: a stochastic local search algorithm (sls) is used to guide the DPLL-like complete solver in choosing literals. That is, every time the solver backtracks because of an occurred conflict, the next literal is chosen with the help of a statistic obtained by running the sls solver.
2. Local search based solvers: a complete solver is used to help out an incomplete one. This can be done in several ways, such as letting DPLL look for dependencies between variables and then limiting the sls solver to a certain subset of variables.
3. Solvers where the incomplete and complete work together but retain their independence: first the local search is run for a certain time, if it fails to find a solution a CDCL solver is executed on the set of currently unsatisfied clauses to find out if they are unsatisfiable.

The original hybrid algorithm described by Mazure et al. [14] belongs to the first category: it uses a local search algorithm to provide a branching heuristic for DPLL. This approach led to a dramatic decrease in the runtime of hard unsatisfiable random instances, especially those that have a locally inconsistent kernel. Their local search algorithm (TSAT, or TWSAT) extends the basic GSAT version from [18, 19] by adding a Tabu list, so that recently flipped variables may not be flipped, preventing the algorithm from getting stuck in local minima and from moving back and forth between a small set of assignments.<sup>3</sup> During the local search, the following trace is recorded: for each literal, it is counted how many times it appears in a falsified clause, taking a flip as a measure of time. Literals that have a high score are more likely to belong to the inconsistent part of the instance than those with low scores. Indeed Mazure et al. discovered that using this literal score as a branching heuristic for a standard DPLL algorithm, whereby TSAT is called every time the DPLL algorithm needs to choose another literal and the literal with the highest score is selected, gives excellent results on locally inconsistent problems.<sup>4</sup> A basic schema of the algorithm is given by Algorithm 1, taken from [14].

In subsequent years, Audemard et al. have also developed other hybrid approaches, belonging to the second category [2, 3]. Nevertheless, the method set forth by the DPLL+TSAT algorithm continues to be pursued and improved upon, as the work of Fourdrinoy demonstrates [8]. In the next section, we explain how we improved this method by making extensive use of parallelism, thereby setting the stage for a new type of hybrid solvers.

<sup>3</sup>Taken by itself, TSAT already proves to be a substantial improvement over the basic GSAT algorithm. See [13].

<sup>4</sup>The popular VSIDS heuristic depends on a somewhat similar scoring mechanism, except that there the focus lies on learned rather than falsified clauses.

## 3 OpenCL-SAT: combining CPU and GPU

### 3.1 Motivation

With the increase of CPU speeds coming to a halt in the last few years, it becomes more and more clear that progress in purely sequential SAT solving can no longer expect to benefit from significant improvements in hardware. Luckily, parallel hardware is becoming ever more widespread, and multicore-CPU's and GPUs can be found in standard PC's. Yet current work on parallel solvers has been limited mostly to the former option, although thanks to the creation of new libraries such as OpenCL it is becoming possible to perform platform-independent GPGPU-computing. The goal of our research is to tap into these unexplored possibilities for SAT solving by designing a solver that exploits parallelism to the fullest.

A naive method would be to simply start from an existing multicore-based parallel approach using four or eight threads, and expanding it to several hundred threads on a GPU. The difference in architecture between both types of hardware makes such an approach impractical:

1. each thread on a GPU has only a very limited amount of (reasonably fast) memory compared to a CPU-core,
2. communication between threads is slow because of the limited bandwidth and is difficult to synchronize,
3. GPUs are made for SIMT (Single Instruction Multiple Threads) algorithms, and thus threads should essentially perform the same algorithm.

One of the main reasons why recent parallel solvers outperform sequential ones, is due to information sharing between threads. Each thread examines one part of the search space, using its own clause database, and passes on clauses to other threads based on their relevance for them. The memory requirements for giving each thread its own database conflict with the first difference and exchanging clauses conflicts with the second. Besides this, the third difference substantially limits the possibilities for threads to perform different tasks concurrently, as can be done on a multicore-CPU.<sup>5</sup>

The fact that all state-of-the-art complete solvers have been developed for the CPU-architecture and thus don't lend themselves to GPU-implementations, is probably the main reason why so little work has been done on complete GPU-based SAT solvers. If we shift our focus to the emerging field of hybrid solvers, however, a new perspective opens up. We can make use of an optimized sequential complete solver on the CPU, while at the same time tap into the enormous processing power offered by the GPU. To implement such a solver, we made use of OpenCL.

### 3.2 OpenCL

GPGPU-programming is still a very young discipline, and until recently CUDA was the only widespread language available to do so. CUDA is a C-based programming language and environment for General Purpose GPU-computing created by NVIDIA. Its syntax and use are quite similar to regular C, except that special attention has to be paid with regard to synchronization for code that will be executed in parallel.

There have been some attempts of implementing massively parallel incomplete solvers on GPUs using CUDA, see for example [16] and [22]. In fact, we used the source code of the latter as a framework to implement our first prototype of a massively parallel incomplete solver.

Since CUDA was designed to run solely on NVIDIA hardware, a group of companies and researchers under the name of the Khronos group decided to develop an open standard, with the intention of providing cross-platform portability. In 2008 this resulted in the conception of OpenCL, or Open Computing Language, an open and royalty free programming framework that enables GPUs, FPGAs, and co-processors to work in tandem with the CPU.

OpenCL is based on ISO C99, with some added extensions and restrictions. The basic unit of executable code that runs on an OpenCL device (a GPU in our case) is called a kernel. The host program is executed on the host system (usually a CPU) and sends kernels to execute on OpenCL devices using a command queue. Besides the programming language, OpenCL also consists of a platform API, which contains routines to

<sup>5</sup>For example, ManySAT's portfolio approach, where each thread can use a completely different solving method, becomes impossible. Also, most parallel solvers use a load-balancing mechanism through work stealing, it is hard to see how this could fit into a SIMT framework.

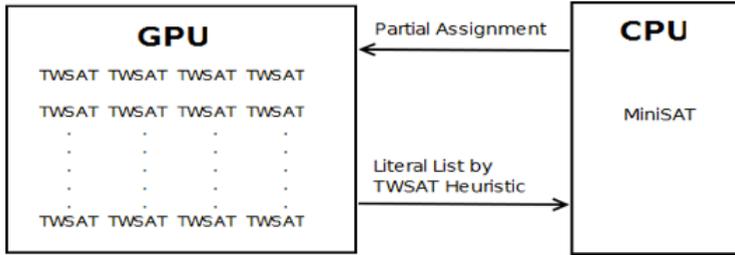


Figure 1: Schema of OpenCL-SAT.

query the system and set up OpenCL resources, and a runtime API, that is used to manage kernel objects, memory objects, and execute kernels on OpenCL devices.

The OpenCL language contains many commands to control the flow of data and the order of execution, thereby giving the programmer a large amount of freedom in specifying the details of how an algorithm should be executed. Therefore it allows one to optimize memory usage and thread synchronization, and offers a tight integration of CPU and GPU execution.

### 3.3 From hybrid to parallel

#### The big picture.

We applied the CPU-GPU perspective to parallelize the DPLL+TSAT technique described in subsection 2.2. The OpenCL framework provides us with the tools for writing programs that combine CPU and GPU resources to suit one’s needs, which fits in nicely with the hybrid character: the CPU is used to run a version of MiniSAT [21], and we use the GPU to run a massively parallel local search that will complement the complete solver. For obvious reasons, we baptized our solver OpenCL-SAT.

In essence, the workings of OpenCL-SAT are rather simple. We adapt MiniSAT so that it no longer uses its VSIDS-like heuristic, but a form of the TSAT heuristic instead. In contrast to previous versions of this hybrid algorithm, we run TSAT in parallel on the GPU at the same time as MiniSAT. (As opposed to the sequential DPLL+TSAT algorithm, where TSAT is called every time MiniSAT needs to choose a literal, so that MiniSAT has to wait for TSAT to finish.) The CPU informs TSAT which variables have been assigned already at every stage, so that future tries of TSAT can run on the partial problem determined by the current subspace in which MiniSAT is looking. Likewise, all literal scores obtained from the local search trace are sent to the CPU, which keeps an ordered list of literals based on this score. Figure 1 depicts the idea behind OpenCL-SAT.

#### The algorithm in detail.

In order for the complete and the incomplete parts to be run concurrently in a sensible way, it is required that the average time for one try of the local search solver is of the same order of magnitude as the time in between two decisions of the complete solver.<sup>6</sup> Yet in the original implementation of the DPLL+TSAT algorithm in [1], one try of the incomplete solver takes as long as 45 decisions of the complete solver.<sup>7</sup>

A straightforward way of dealing with this is to use the literal score obtained from one try for 45 decisions of the complete solver. Doing this however results in a dramatic decrease of the effectiveness of the heuristics, making it all but useless. The only viable option is to drastically speed-up the calculation of the heuristic, without compromising its quality too much. We succeeded in this by putting to use the massive parallelism offered by a GPU. The GPU is used to obtain a heuristic which is based on the original idea - namely calculating a score for each literal by counting the number of its appearances in falsified clauses, but modified in several ways so as to function optimally on the chosen type of hardware.

<sup>6</sup>A stochastic local search algorithm consists in several *tries*, where each try is a certain amount of *flips* being performed on a random initial assignment.

<sup>7</sup>We obtained this number by running the original SUN-solver from [12] on the same test set of random instances that we also used in section 4.

---

**Algorithm 2** basic local search solver

---

**Procedure** *local search kernel function*

**Input** : a set of clauses  $S$  over a set of variables  $V$ , an initial assignment  $A$  to  $V$ ,  $\text{MAX-ITER}$ ,  $i_0 < |V|$

**Output** : an assignment to  $V$  satisfying  $S$ , if found  
or a score for all literals from  $V$  if not

**Begin**

$i = i_0$ ;  $PreviousChoice = i_0$ ;

**while**  $((i < \text{MAX-ITER} + i_0) \wedge (i - PreviousChoice) < |V|)$

$v = i \bmod |V|$ ;

**if**  $AlreadyAssignedMiniSAT(v)$  **then**  $(i++)$ ; *continue*

**if**  $FalseClauses(S, A + flip(v)) < FalseClauses(S, A)$  **then**

$A = A + flip(v)$ ;

$PreviousChoice = v$ ;

$UpdateScore(A)$ ;

**if**  $FalseClauses(A) = 0$  **then return** "satisfiable";

**else**  $i++$ ;

**end while**;

**return** "no satisfying assignment found";

**End.**

---

More specifically, we ran a large number of threads of a very basic stochastic local search solver depicted schematically in Algorithm 2.<sup>8</sup> The TSAT score for a literal is then taken to be the average of the scores given by each thread.

In a first attempt each thread started out with a different initial assignment, and ran until either a local minimum was reached or some maximum amount of iterations had occurred. We noticed two things:

1. even our very basic local search algorithm couldn't meet the speed requirements;
2. the quality of the heuristic didn't improve notably beyond using 200 threads.

So we decided to group several threads together into about 200 groups, and tried to let each group mimic the behavior of one independent try of the sls solver. We did this by letting each thread in a group start off with the same (random) initial assignment, and then flipped one randomly chosen variable for each thread. (Where we made sure that no two threads flipped the same variable, i.e. the assignments for two threads in a group differ in exactly two places.) Then we ran our basic local search algorithm, where we reduced the number of iterations by a factor of 100.<sup>9</sup> The idea is that assignments that are close to each other - measured by the Hamming distance - represent different iterations of a single initial assignment undergoing the local search algorithm. By using this strategy, we were able to bring the average time of one try of the incomplete solver down to around the same order of magnitude as an average decision by MiniSAT, and still succeeded in mimicking the original heuristic obtained in our first attempt.

There are several ways in which we had to adapt the TSAT local search algorithm in order to fit into this framework. Because of the very limited amount of available local memory on a GPU per thread, there was no space to remember useful statistics, such as a list of all clauses a certain variable appears in. Therefore every thread simply goes over all variables, and flips it if this decreases the number of falsified clauses in the current assignment. Variables that have already been assigned a value by MiniSAT are of course excluded from this process. After every flip we check whether all clauses are satisfied, if not then the literal score is updated. Most random walk search algorithms integrate random flips every now and then, but since we are averaging over a large number of threads we did not. Also, the use of a Tabu list<sup>10</sup> was left out of our algorithm for two reasons: firstly, in our set up it is impossible for the same variable to be flipped twice successively; secondly, it is less likely that a recently flipped variable is being considered to be flipped again.

---

<sup>8</sup>Depending on the GPU this can go up to 6000, for an NVIDIA Tesla GPU, and around 800 for a more standard NVIDIA Quadro FX2700M.

<sup>9</sup>The number of iterations was reduced to  $0.02 * v$ , where  $v$  stands for the number of variables that are currently unassigned by MiniSAT.

<sup>10</sup>I.e. a list of recently flipped variables that may not be chosen, in order to avoid getting stuck in local minima.

	OpenCL-SAT	DPLL+TSAT	MiniSAT
TOTAL RUNTIME	175.97	2, 431.31	48.20
TOTAL DECISIONS	866, 898	432, 389	418, 908

Figure 2: Results.

## 4 Results

We have tested the OpenCL-SAT solver on 50 random 3-SAT instances from the DIMACS SAT library [23], where we ran each instance with four different random seeds. We then took the average over all problems of the same size. The sizes ranged from 150 to 250 variables, with a clauses/variables ratio of 4.26 for all instances, meaning that they lie in the hardest region. We chose to focus on hard random instances because they are significantly smaller than crafted or industrial ones of comparable difficulty, small enough in fact to fit into local memory of the GPU, which makes the solver significantly faster than if it were using global memory.

We compared OpenCL-SAT to the original hybrid DPLL+TSAT solver, as well as to MiniSAT using its standard heuristics.<sup>11</sup> The tests for OpenCL-SAT and MiniSAT were run on an NVIDIA Tesla C2075 GPU and a Intel Xeon E5645 CPU.<sup>12</sup> In total there were 5600 threads running on the GPU, divided into workgroups of 20x20.<sup>13</sup>

Figure 2 shows the total runtimes and number of decisions.<sup>14</sup> First of all, we can observe that OpenCL-SAT is about 13 times faster than the DPLL+TSAT solver, confirming the possible advantages of the CPU-GPU framework. However, there is still room for improvement concerning the quality of the heuristics, as the DPLL+TSAT solver required about only half the number of decisions. It should be possible to divide the workload more efficiently between all threads, making the implementation more scalable. Furthermore, it's clear that for random instances the current approach cannot yet compete with MiniSAT concerning neither the quality of the heuristic nor the runtimes, even when we take into account the number of decisions.

## 5 Conclusions and Future Work

We have built a SAT solver that is both parallel and hybrid. By doing so we have combined two trends in the SAT solving community that are becoming widespread. We were able to do so by making use of OpenCL, that taps into the resources offered by combined CPU-GPU computing. Our solver was based on the fairly basic hybrid algorithm from [14], but more complex hybrid algorithms have since been developed that perform better on a wide set of instances (see [15] for an overview). In the future, similar solvers can be developed that implement these complex algorithms. The challenge will be to adapt them in such a way that they can benefit optimally from the GPU's parallelism, without suffering too much from the GPU's limitations. In the long run, however, attention should shift to algorithms that are designed from the bottom-up with a CPU-GPU set up in mind. To conclude, although our solver is as yet still too basic to compete with state-of-the-art sequential solvers, it is a first step in the direction of a new type of solvers that contain a lot of potential.

## References

- [1] Audemard, G., Lagniez, J., Mazure, B. and Saïs, L.: Boosting local search techniques thanks to CDCL. In: 17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, pp. 474–488 (2010)

<sup>11</sup>We did not make use of any restart policy, since in general only industrial and crafted instances benefit from this.

<sup>12</sup>For the DPLL+TSAT solver we used an Intel T9600 CPU, which makes the comparison of runtimes a bit unfair. However, when compared to the runtimes of our solver on this same CPU together with a more basic GPU, the results weren't that different. See the following footnote.

<sup>13</sup>We used a more modest NVIDIA Quadro FX2700M together with an Intel T9600 as well, running 864 threads. The total runtime increased only by 15%, so unfortunately in its present form the solver isn't very scalable. The main reason for this is that although adding more threads can greatly speed up the calculation of the heuristics to be used by MiniSAT, it has little influence on its quality.

<sup>14</sup>For each problem size we took the average for both runtime and decisions.

- [2] Audemard, G., Lagniez, J., Mazure, B. and Saïs, L.: Integrating Conflict Driven Clause Learning to Local Search. In: 6th International Workshop on Local Search Techniques in Constraint Satisfaction, pp. 55–68, (2009)
- [3] Audemard, G., Lagniez, J., Mazure, B. and Saïs, L.: Learning in local search. In: 21st IEEE International Conference on Tools with Artificial Intelligence, pp. 417–424 (2009)
- [4] Beckers, S., De Samblanx, G., De Smedt, F., Goedemé, T., Struyf, L. and Vennekens, J.: Parallel SAT-solving with OpenCL. In: IADIS International Conference on Applied Computing, pp. 435–440 (2011)
- [5] Biere, A.: Lingeling, Plingeling, PicoSAT and PrecoSAT at SAT Race 2010. Technical Report, FMV Reports Series (2010)
- [6] Chu, G. and Stuckey, P.J.: PMiniSAT: a parallelization of MiniSAT 2.0. In: SAT race (2008)
- [7] Davis M., Putnam H.: A Computing Procedure for Quantification Theory. *Journal of the ACM*, 7, 201–215 (1960)
- [8] Fourdrinoy, O.: Utilisation de techniques polynomiales pour la résolution pratique d’instances de SAT. Thèse de doctorat. (2007)
- [9] Hamadi, Y.; Saïs, L.: ManySAT: a parallel SAT solver. *Journal on Satisfiability, Boolean Modelling and Computation* (2009)
- [10] Lewis, M., Schubert, T. and Becker B.: Multithreaded SAT Solving. In: 12th Asia and South Pacific Design Automation Conference, pp. 926–931 (2007)
- [11] Mazure B., Saïs, L., and Grégoire E.: TWSAT : a new local search algorithm for SAT : performance and analysis. In: the Workshop CP95 on Solving Really Hard Problems, pp. 127–130 (1995)
- [12] Mazure B., Saïs, L., and Grégoire E.: SUN : a Multistrategy Platform for SAT. In: First International Competition and Symposium on Satisfiability Testing, SAT solvers description (1996)
- [13] Mazure B., Saïs, L., and Grégoire E.: Tabu search for sat. In: 14th National Conference on Artificial Intelligence, pp. 281–285 (1997)
- [14] Mazure B., Saïs, L., and Grégoire E.: Boosting complete techniques thanks to local search methods. *Annals of Mathematics and Artificial Intelligence*, 22, 319–331 (1998)
- [15] Mazure, B.: SAT et au-delà de SAT : Modèles et Algorithmes. Habilitation à Diriger des Recherches (2010)
- [16] Meyer, Q., Schonfeld, F., Stamminger, M. and Wanka R.: 3-SAT on CUDA: Towards a massively parallel SAT solver. In: High Performance Computing Symposium, pp.306–313 (2010)
- [17] Moskewicz, M., Madigan, C., Zhao, Y., Zhang, L. and Maliket, S.: Chaff: Engineering an Efficient SAT solver. In: the Design Automation Conference, pp. 530–535 (2001)
- [18] Selman, B., Levesque, H. and Mitchell, D.: A New Method for Solving Hard Satisfiability Problems. In: 9th National Conference on Artificial Intelligence, pp. 440–446 (1992)
- [19] Selman, B., Kautz, H. and Cohen, B.: Local Search Strategies for Satisfiability Testing. In: Working notes of the DIMACS Workshop on Maximum Clique, Graph Coloring, and Satisfiability (1993)
- [20] Selman, B., Kautz, H. and McAllister, D.: Ten Challenges in Propositional Reasoning and Search. In: 15th International Joint Conference on Artificial Intelligence, vol. 1, pp. 50–54 (1997)
- [21] Sörensson, N; Eén, N.: An Extensible SAT-solver. *Lecture Notes in Computer Science*, 2912, 333–336 (2004)
- [22] Wang, Y.: NVIDIA CUDA Architecture-based Parallel Incomplete SAT solver. Master Project Final Report, Faculty of Rochester Institute of Technology (2010)
- [23] <http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html>

# A Proof of the Expression of the Prior Convergence Error

Janneke H. Bolt

*Department of Information and Computing Sciences, Utrecht University  
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands  
janneke@cs.uu.nl*

## Abstract

In [2], we introduced the notion of the *parental synergy*. In the same paper, moreover, an expression was advanced for the prior convergence error (the error which is found in the marginal probabilities computed for a node when the parents of this node are wrongfully assumed to be independent), in which the parental synergy has a key position as weighting factor. This key position suggests that the parental synergy captures a fundamental feature of a Bayesian network. In this paper we provide a proof for the correctness of the conjectured expression of the prior convergence error.

## 1 Introduction

A Bayesian network is a concise representation of a joint probability distribution over a set of stochastic variables, consisting of a directed acyclic graph and a set of conditional probability distributions [3]. The nodes of the graph represent the variables of the distribution and the arcs of the graph capture (conditional) independencies. From a Bayesian network, in theory, any probability of the represented distribution can be inferred. In the computation of the marginal probabilities of a node, the dependencies between its parent nodes have to be taken into account. When these dependencies are neglected, an error may arise. In [1], we termed this error the prior convergence error.

In [2] we introduced the notion of the parental synergy. The parental synergies of a node are computed from the parameters as specified for this node in a Bayesian network. In the same paper, we conjectured an expression for the prior convergence error for the general case of a child node with an arbitrary number of dependent parent nodes. The proposed expression is of interest because of its structure. It is composed of a part that captures the degree of the dependency between the parent nodes, and of the parental synergies of the node. In the expression of the prior convergence error the parental synergies act as weighting factors, determining to what extent the degree of dependency between the parent nodes can affect the computed probabilities. We stated that the role of the parental synergy in the expression of the prior convergence error suggests that it captures a fundamental feature of a Bayesian network. In this respect, we note that the parental synergy is related to the concepts of qualitative influence and additive synergy as defined for qualitative probabilistic networks by Wellman [4]. The concept of parental synergy, however, is more general and more informative. In this paper, we provide a proof of the correctness of the expression of the prior convergence error that we conjectured in [2].

## 2 General Preliminaries

We will use the following notation: Variables are denoted by upper-case letters ( $A, A_k$ ), and their values by lower-case letters ( $a_i, a_{i_k}$ ); sets of variables by bold-face upper-case letters ( $\mathbf{A}, \mathbf{A}_k$ ) and their instantiations by bold-face lower-case letters ( $\mathbf{a}_i, \mathbf{a}_{i_k}$ ). An arbitrary value assignment to  $A$  may also be indicated by  $a$  instead of  $a_i$  and an arbitrary joint value assignment to  $\mathbf{A}$  may also be indicated by  $\mathbf{a}$  instead of  $\mathbf{a}_i$ . The upper-case letter is also used to indicate the whole range of values or value combinations of a variable or a set of variables.

Figure 1 depicts the graph of a Bayesian network. The network includes a node  $C$  with  $n$ , possibly dependent, parents  $\mathbf{A} = A_1, \dots, A_n$ ,  $n \geq 0$ . In this network, the marginal probability  $\Pr(c)$ , of an arbitrary value  $c$  of  $C$ , equals

$$\Pr(c) = \sum_{\mathbf{A}} \Pr(c \mid \mathbf{A}) \cdot \Pr(\mathbf{A})$$

Wrongfully assuming independence of the parents  $\mathbf{A}$  would yield the approximation

$$\widetilde{\Pr}(c) = \sum_{\mathbf{A}} \Pr(c \mid \mathbf{A}) \cdot \Pr(A_1) \cdot \dots \cdot \Pr(A_n)$$

In [1], we defined the prior convergence to equal  $\Pr(c) - \widetilde{\Pr}(c)$ , and in [2], we conjectured an expression for the prior convergence error given a node with an arbitrary number of arbitrary-valued, possibly dependent, parent nodes, as depicted in Figure 1. We restate this expression below.

The parental synergy, a notion that we introduced in [2], is an important factor in the conjectured expression of the prior convergence error and in the definition of the parental synergy an indicator function, called  $\delta$ , is used. The definitions of this indicator function and of the parental synergy are stated first.

**Definition 1. (the indicator function  $\delta$ )**

Let  $\mathbf{A}$  and  $\mathbf{B}$  be disjoint sets of variables. The indicator function  $\delta$  on the joint value assignments  $a_{i_1}, \dots, a_{i_n}$  to the set of variables  $A = A_1, \dots, A_n$ ,  $n \geq 0$ , given a specific assignment  $a_{s_1}, \dots, a_{s_n}$  and an arbitrary value assignments  $\mathbf{b}$  is:

$$\delta(a_{i_1}, \dots, a_{i_n} \mid a_{s_1}, \dots, a_{s_n} \mathbf{b}) = \begin{cases} 1 & \text{if } \sum_{k=1, \dots, n} a_{i_k} \neq a_{s_k} \text{ is even} \\ -1 & \text{if } \sum_{k=1, \dots, n} a_{i_k} \neq a_{s_k} \text{ is odd} \end{cases}$$

where true  $\equiv 1$  and false  $\equiv 0$ . □

The indicator function compares the joint value assignment  $a_{i_1}, \dots, a_{i_n}$  with the assignment  $a_{s_1}, \dots, a_{s_n}$ , and counts the number of differences: the assignment  $a_{i_1}, \dots, a_{i_n}$  is mapped to the value 1 if the number of differences is even and is mapped to  $-1$  if the number of differences is odd. We note that although  $\mathbf{b}$  has no influence on the outcome of  $\delta$ , for notational reasons it is convenient if the function allows for this arbitrary value assignment.

**Definition 2. (the parental synergy)**

Let  $\mathcal{B}$  be a Bayesian network, representing a joint probability distribution  $\Pr$  over a set of variables  $\mathbf{V}$ . Let  $\mathbf{A} = \{A_1, \dots, A_n\} \subseteq \mathbf{V}$ ,  $n \geq 0$ , and let  $C \in \mathbf{V}$  such that  $C$  is a child of all variables in the set  $\mathbf{A}$ , that is,  $A_j \rightarrow C$ ,  $j = 1, \dots, n$ . Let  $\mathbf{a}$  be a joint value assignment to  $\mathbf{A}$  and let  $c$  be a value of  $C$ . Furthermore, let  $\mathbf{X} \subseteq \rho(C) \setminus \mathbf{A}$ , where  $\rho(C)$  denotes the parents of  $C$ , and let  $\mathbf{x}$  be a value assignment to  $\mathbf{X}$ . The *parental synergy* of  $\mathbf{a}$  with respect to  $c$  given  $\mathbf{X} = \mathbf{x}$ , denoted as  $Y_{\mathbf{x}}(c \mid \mathbf{a})$ , is

$$\begin{aligned} Y_{\mathbf{x}}(c \mid \mathbf{a}) &= \sum_{\mathbf{A}} \delta(\mathbf{a} \mid \mathbf{A}) \cdot \Pr(c \mid \mathbf{A} \mathbf{x}) \\ Y_{\mathbf{x}}(c) &= \Pr(c \mid \mathbf{x}) \end{aligned} \quad \square$$

*Example 1.* For a node  $C$  with parents  $A$  with values  $a_1, a_2$  and  $a_3$  and  $B$  with values  $b_1$  and  $b_2$ , with  $\Pr(c \mid a_1 b_1) = r$ ,  $\Pr(c \mid a_1 b_2) = s$ ,  $\Pr(c \mid a_2 b_1) = t$ ,  $\Pr(c \mid a_2 b_2) = u$ ,  $\Pr(c \mid a_3 b_1) = v$  and  $\Pr(c \mid a_3 b_2) = w$ ,  $Y(c \mid a_1 b_1) = r - s - t + u - v + w$ ,  $Y_{a_2}(c \mid b_2) = -t + u$  and  $Y_{a_1 b_2}(c) = s$ . □

Note that the parental synergy is related to the concepts of qualitative influence and additive synergy as defined for qualitative probabilistic networks by Wellman [4]. Most obviously, in a binary network, given a node  $C$  with a single parent  $A$ , the sign of the qualitative influence between  $A$  and  $C$  is derived from  $\Pr(c \mid a) - \Pr(c \mid \bar{a})$ , which equals  $Y(c \mid a)$ ; given a node  $C$  with just the parents  $A$  and  $B$  the sign of the additive synergy of  $A$  and  $B$  with respect to  $C$  is derived from  $\Pr(c \mid ab) - \Pr(c \mid a\bar{b}) - \Pr(c \mid \bar{a}b) + \Pr(c \mid \bar{a}\bar{b})$ , which equals  $Y(c \mid ab)$ . The parental synergy, however is more general since it is defined for an arbitrary

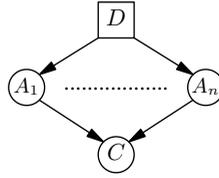


Figure 1: A graph of a Bayesian network with a node  $C$  with the, possibly dependent, parents  $A_1, \dots, A_n$ .

number of parent nodes whereas the qualitative influence concerns the interaction between a child node and one parent node, the additive synergy concerns the interactions between a child node and two parent nodes and no analogous concepts are defined for interactions between a child node and more than two parent nodes. Moreover, the parental synergy is more informative, since it yields a number whereas the qualitative influence and the additive synergy are given by a '+' , a '-' or the uninformative sign '?' .

In [2], we conjectured an expression for prior convergence error  $\Pr(c) - \widetilde{\Pr}(c)$ . This expression is stated in the following theorem.

**Theorem 1.**

Let  $\mathcal{B}$  be a Bayesian network, representing a joint probability distribution  $\Pr$  over a set of arbitrary-valued variables  $\mathbf{V}$ . Let  $C \in \mathbf{V}$  and let  $\rho(C) = \mathbf{A} = \{A_1, \dots, A_n\} \subseteq \mathbf{V}, n \geq 0$  be the set of, possibly dependent, parents of  $C$ . The prior convergence error  $\Pr(c) - \widetilde{\Pr}(c)$  then equals

$$\Pr(c) - \widetilde{\Pr}(c) = \sum_{\mathbf{m}} \left[ \sum_{\mathbf{A}_{\mathbf{m}}} \left( \Pr(A_x \dots A_y) - \Pr(A_x) \cdot \dots \cdot \Pr(A_y) \right) \cdot \sum_{\mathbf{A} \setminus \mathbf{A}_{\mathbf{m}}} Y_{\mathbf{A} \setminus \mathbf{A}_{\mathbf{m}}}(c \mid \mathbf{A}_{\mathbf{m}}) \right] / 2^n$$

where

$$\begin{aligned} \{\mathbf{m}\} &= \mathcal{P}(\{1, \dots, n\}) \\ \mathbf{A}_{\mathbf{m}} &= \{A_x \dots A_y\}, \mathbf{m} = \{x, \dots, y\} \end{aligned} \quad \square$$

By the summation over  $\mathbf{m}$ , all combinations of parent nodes are considered, moreover, by the summation over  $\mathbf{A}_{\mathbf{m}}$  for each combination of parent nodes all combinations of value assignments to these parent nodes are taken into account. In the remainder of the paper, we will keep using these notations  $\mathbf{m}$  and  $\mathbf{A}_{\mathbf{m}}$ .

The expression above is very complex and its value therefore is mainly theoretical. The expression shows that the parental synergy is a weighting factor that determines the impact of the degree of dependency between the parent nodes for a given value assignment, as reflected by  $\Pr(A_x \dots A_y) - \Pr(A_x) \cdot \dots \cdot \Pr(A_y)$  on the size of the convergence error. Note that if the number of elements of  $\mathbf{m}$  is smaller than two, that is, if just one parent or zero parents are considered, then  $\Pr(A_x \dots A_y)$  equals  $\Pr(A_x) \cdot \dots \cdot \Pr(A_y)$  and thus  $\Pr(A_x \dots A_y) - \Pr(A_x) \cdot \dots \cdot \Pr(A_y)$  equals zero.

### 3 The Prior Convergence Error Illustrated

Given a node with just two, binary, parent nodes, the prior convergence error and the key position of the parental synergy can be illuminated graphically. Consider Figure 1 and consider that  $n = 2$ , that  $A_1$  and  $A_2$  are binary. The expression of the prior convergence error then reduces to

$$\begin{aligned} \Pr(c) - \widetilde{\Pr}(c) &= \left[ \sum_{A_1 A_2} (\Pr(A_1 A_2) - \Pr(A_1) \cdot \Pr(A_2)) \cdot Y(c \mid A_1 A_2) \right] / 4 \\ &= (\Pr(a_1 a_2) - \Pr(a_1) \cdot \Pr(a_2)) \cdot Y(c \mid a_1 a_2) \end{aligned}$$

which can be rewritten as

$$\Pr(c) - \widetilde{\Pr}(c) = l \cdot m \cdot n \cdot Y(c | a_1 a_2)$$

where

$$\begin{aligned} l &= \Pr(d) - \Pr(d)^2 \\ m &= \Pr(a_1 | d) - \Pr(a_1 | \bar{d}) \\ n &= \Pr(a_2 | d) - \Pr(a_2 | \bar{d}) \end{aligned}$$

Now consider for example that  $\Pr(d) = 0.5$ ,  $\Pr(a_1 | d) = 0.5$ ,  $\Pr(a_1 | \bar{d}) = 0.9$ ,  $\Pr(a_2 | d) = 0.1$ ,  $\Pr(a_2 | \bar{d}) = 0.9$ ,  $\Pr(c | a_1 a_2) = 1$ ,  $\Pr(c | a_1 \bar{a}_2) = 0$ ,  $\Pr(c | \bar{a}_1 a_2) = 0$  and  $\Pr(c | \bar{a}_1 \bar{a}_2) = 1$ . The prior convergence error for this example is illustrated in Figure 2(a). The line segment in this figure captures the exact probability  $\Pr(c)$  as a function of  $\Pr(d)$ .  $\Pr(d)$  itself is not indicated in the figure, note however, that each particular  $\Pr(d)$  has a corresponding  $\Pr(a_1)$  and  $\Pr(a_2)$ . The end points of the line segment, for example, are found at  $\Pr(d) = 1$  with the corresponding  $\Pr(a_1) = 0.5$  and  $\Pr(a_2) = 0.1$  and at  $\Pr(d) = 0$  with the corresponding  $\Pr(a_1) = 0.9$  and  $\Pr(a_2) = 0.9$ . The surface captures  $\widetilde{\Pr}(c)$  as a function of  $\Pr(a_1)$  and  $\Pr(a_2)$ . The convergence error equals the distance between the point on the line segment that matches the probability  $\Pr(d)$  from the network and its orthogonal projection on the surface. For  $\Pr(d) = 0.5$  the difference between  $\Pr(c)$  and  $\widetilde{\Pr}(c)$  is indicated by the vertical dotted line segment and equals  $0.66 - 0.5 = 0.16$ . The factor  $l$  reflects the location of the point with the exact probability on the line segment and the factors  $m$  and  $n$  reflect the location of the point of the line segment. The parental synergy  $Y(c | a_1 a_2)$  now reflects the curvature of the surface with the approximate probabilities. The curvature of the surface determines to what extent the dependency between  $A_1$  and  $A_2$  can affect the computed probabilities. In the example, the curvature of the surface is maximal. Figure 2(b) shows, in contrast, a situation in which the parental synergy equals zero. In this example the specifications for nodes  $D$ ,  $A_1$  and  $A_2$  remained the same, but the specification for node  $C$  has changed to  $\Pr(c | a_1 a_2) = 0.6$ ,  $\Pr(c | a_1 \bar{a}_2) = 0.1$ ,  $\Pr(c | \bar{a}_1 a_2) = 1.0$  and  $\Pr(c | \bar{a}_1 \bar{a}_2) = 0.5$ . Now, the surface is flat;  $Y(c | a_1 a_2) = 0$  and the prior convergence error equals zero.

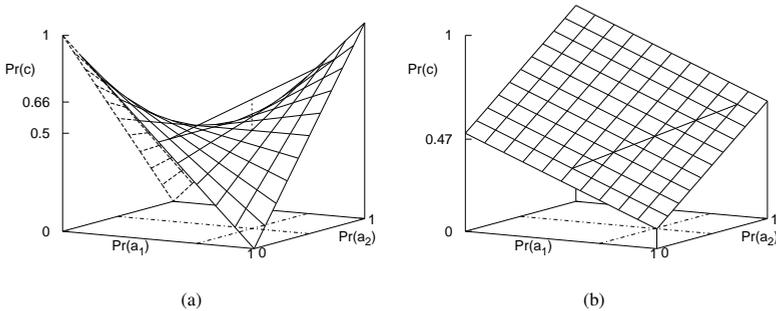


Figure 2: The prior convergence error  $\Pr(c) - \widetilde{\Pr}(c)$  illustrated, given a child node with just two, binary, parent nodes  $A_1$  and  $A_2$ , given  $Y(c | ab) = 2$  (a) and given  $Y(c | ab) = 0$  (b).

## 4 A Proof the Expression of the Prior Convergence Error

In order to proof the validity of Theorem 1 we propose the following expressions for  $\Pr(c)$  and  $\widetilde{\Pr}(c)$ .

### Proposition 1.

Let  $\mathcal{B}$  be a Bayesian network, representing a joint probability distribution  $\Pr$  over a set of variables  $\mathbf{V}$ . Let  $C \in \mathbf{V}$  and let  $\rho(C) = \mathbf{A} = \{A_1, \dots, A_n\} \subseteq \mathbf{V}$ ,  $n \geq 0$  be the set of, possibly dependent, parents of  $C$ . The prior probability  $\Pr(c)$  then equals

$$\Pr(c) = \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \sum_{\mathbf{A} \setminus \mathbf{A}_m} Y_{\mathbf{A} \setminus \mathbf{A}_m}(c | \mathbf{A}_m) / 2^n \quad (1)$$

*Example 2. For a variable  $C$  with the parents  $A_1$  and  $A_2$ , according to Proposition 1*

$$4 \cdot \Pr(c) = \sum_{A_1 A_2} \Pr(A_1 A_2) \cdot Y(c | A_1 A_2) + \sum_{A_1} \Pr(A_1) \cdot \sum_{A_2} Y_{A_2}(c | A_1) + \sum_{A_2} \Pr(A_2) \cdot \sum_{A_1} Y_{A_1}(c | A_2) + \sum_{A_1 A_2} Y_{A_1 A_2}(c) \quad \square$$

Given Proposition 1, the approximation  $\widetilde{\Pr}(c)$  can be written as:

$$\widetilde{\Pr}(c) = \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(A_x) \cdot \dots \cdot \Pr(A_y) \cdot \sum_{\mathbf{A} \setminus \mathbf{A}_m} Y_{\mathbf{A} \setminus \mathbf{A}_m}(c | \mathbf{A}_m) / 2^n \quad (2)$$

In the proof of Proposition 1 we will use the following lemma:

**Lemma 1.**

Let  $\mathbf{A} = \{A_1, \dots, A_n\}$  be a set of variables and let  $\mathbf{a} = a_1, \dots, a_n$  be an arbitrary given joint value assignment to  $\mathbf{A}$ . Then  $2^n \cdot \Pr(\mathbf{a})$  equals

$$2^n \cdot \Pr(\mathbf{a}) = \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \delta(\mathbf{A}_m | \mathbf{a}) \quad \square$$

*Example 3. Given the variables  $A$  with the values  $a$  and  $\bar{a}$  and  $B$  with the values  $b$  and  $\bar{b}$ , according to Lemma 1,  $4 \cdot \Pr(ab) = \Pr(ab) - \Pr(a\bar{b}) - \Pr(\bar{a}b) + \Pr(\bar{a}\bar{b}) + \Pr(a) - \Pr(\bar{a}) + \Pr(b) - \Pr(\bar{b}) + 1$   $\square$*

**Proof of Lemma 1.**

We first rewrite

$$\begin{aligned} 2^n \cdot \Pr(\mathbf{a}) &= \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \delta(\mathbf{A}_m | \mathbf{a}) \\ &= \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \sum_{\mathbf{A} \setminus \mathbf{A}_m} \Pr(\mathbf{A} \setminus \mathbf{A}_m, \mathbf{A}_m) \cdot \delta(\mathbf{A}_m | \mathbf{a}) \\ &= \sum_{\mathbf{m}} \sum_{\mathbf{A}} \Pr(\mathbf{A}) \cdot \delta(\mathbf{A}_m | \mathbf{a}) \end{aligned}$$

In the expression  $\sum_{\mathbf{m}} \sum_{\mathbf{A}} \Pr(\mathbf{A}) \cdot \delta(\mathbf{A}_m | \mathbf{a})$  we have that  $\sum_{\mathbf{m}}$  selects all possible combinations of the variables of  $\mathbf{A}$ , and  $\sum_{\mathbf{A}}$  sums, for each of those combinations,  $\Pr(\mathbf{A}) \cdot \delta(\mathbf{A}_m | \mathbf{a})$ . The sign of the outcome of the function  $\delta$  is determined by the variables selected by  $\mathbf{m}$  with the values as determined by  $\sum_{\mathbf{A}}$ , and the values of  $\mathbf{A}$  in the given joint value assignment  $\mathbf{a}$ .

*Example 4. Given the variables  $A$  and  $B$  and the value assignment  $A = a$  and  $B = b$ , the expression*

$$\sum_{\mathbf{m}} \sum_{\mathbf{A}} \Pr(\mathbf{A}) \cdot \delta(\mathbf{A}_m | \mathbf{a})$$

results in:

$$\sum_{AB} \Pr(AB) \cdot \delta(AB | ab) + \sum_{AB} \Pr(AB) \cdot \delta(A | ab) + \sum_{AB} \Pr(AB) \cdot \delta(B | ab) + \sum_{AB} \Pr(AB) \cdot \delta(\cdot | ab)$$

$\square$

Now divide  $\{\mathbf{m}\}$  in  $\{\mathbf{m}_{-1}\} = \{\mathbf{m} | 1 \notin \mathbf{m}\}$  and  $\{\mathbf{m}_{+1}\} = \{\mathbf{m} | 1 \in \mathbf{m}\}$ . Thus,  $\mathbf{m}_{-1}$  selects all possible subsets from  $\mathbf{A}$  in which  $A_1$  is included and  $\mathbf{m}_{+1}$  selects all possible subsets from  $\mathbf{A}$  without  $A_1$ . These two sets of subsets of  $\mathbf{A}$  include, in pairs, the same subsets, apart from  $A_1$ . Thus,  $\emptyset$  is selected by  $\mathbf{m}_{-1}$  and

$\{A_1\}$  is selected by  $\mathbf{m}_{+1}$ ;  $\{A_2\}$  is selected by  $\mathbf{m}_{-1}$  and  $\{A_1, A_2\}$  is selected by  $\mathbf{m}_{+1}$ , etcetera. For each of those pairs, we find for all value combinations  $\mathbf{A}$  with  $A_1 \neq a_1$  that

$$\delta(\mathbf{A}_{\mathbf{m}_{-1}} | \mathbf{a}) = -\delta(\mathbf{A}_{\mathbf{m}_{+1}} | \mathbf{a})$$

(Remember that  $\mathbf{a} = a_1, \dots, a_n$ . The outcome of  $\delta(\mathbf{A}_{\mathbf{m}_{-1}} | \mathbf{a})$  is, apart from  $A_1$ , determined by the same value assignments to the same variables as the outcome of  $\delta(\mathbf{A}_{\mathbf{m}_{+1}} | \mathbf{a})$ . Now when  $A_1 \neq a_1$  there is one extra difference in value assignment, which changes the sign of the outcome of  $\delta$ .) Thus under the condition that  $A_1 \neq a_1$  we find that

$$\sum_{\mathbf{m}_{+1}} \sum_{\mathbf{A}} \Pr(\mathbf{A}) \cdot \delta(\mathbf{A}_{\mathbf{m}_{+1}} | \mathbf{a}) + \sum_{\mathbf{m}_{-1}} \sum_{\mathbf{A}} \Pr(\mathbf{A}) \cdot \delta(\mathbf{A}_{\mathbf{m}_{-1}} | \mathbf{a}) = 0$$

This implies that we only have to consider  $A_1 = a_1$  and thus that

$$\begin{aligned} 2^n \cdot \Pr(\mathbf{a}) &= \sum_{\mathbf{m}} \sum_{\mathbf{A}} \Pr(\mathbf{A}) \cdot \delta(\mathbf{A}_{\mathbf{m}} | \mathbf{a}) \\ &= \sum_{\mathbf{m}_{+1}} \sum_{\mathbf{A}} \Pr(\mathbf{A}) \cdot \delta(\mathbf{A}_{\mathbf{m}_{+1}} | \mathbf{a}) + \sum_{\mathbf{m}_{-1}} \sum_{\mathbf{A}} \Pr(\mathbf{A}) \cdot \delta(\mathbf{A}_{\mathbf{m}_{-1}} | \mathbf{a}) \\ &= \sum_{\mathbf{m}} \sum_{\mathbf{A}/A_1} \Pr(\mathbf{A}/A_1, a_1) \cdot \delta(\mathbf{A}_{\mathbf{m}}(/A_1, a_1) | \mathbf{a}) \end{aligned}$$

where

$$\delta(\mathbf{A}_{\mathbf{m}}(/A_1, a_1) | \mathbf{a}) = \begin{cases} \delta(\mathbf{A}_{\mathbf{m}}/A_1, a_1 | \mathbf{a}) & \text{if } 1 \in \mathbf{m} \\ \delta(\mathbf{A}_{\mathbf{m}} | \mathbf{a}) & \text{if } 1 \notin \mathbf{m} \end{cases}$$

In a next step we divide  $\mathbf{m}$  in  $\mathbf{m}_{-2}$  and  $\mathbf{m}$  in  $\mathbf{m}_{+2}$ . We have, analogous to the division of  $\mathbf{m}$  in  $\mathbf{m}_{-1}$  and  $\mathbf{m}$  in  $\mathbf{m}_{+1}$ , that for all value combinations  $\mathbf{A}/A_1$ , given that  $A_2 \neq a_2$

$$\delta(\mathbf{A}_{\mathbf{m}_{-2}}(/A_1, a_1) | \mathbf{a}) = -\delta(\mathbf{A}_{\mathbf{m}_{+2}}(/A_1, a_1) | \mathbf{a})$$

And thus we have that:

$$\begin{aligned} 2^n \cdot \Pr(\mathbf{a}) &= \sum_{\mathbf{m}} \sum_{\mathbf{A}/A_1} \Pr(\mathbf{A}/A_1, a_1) \cdot \delta(\mathbf{A}_{\mathbf{m}}(/A_1, a_1) | \mathbf{a}) \\ &= \sum_{\mathbf{m}} \sum_{\mathbf{A}/A_1 A_2} \Pr(\mathbf{A}/A_1 A_2, a_1 a_2) \cdot \delta(\mathbf{A}_{\mathbf{m}}(/A_1 A_2, a_1 a_2) | \mathbf{a}) \end{aligned}$$

This pinning down of the value of a variable in the summation over  $\mathbf{A}$  can be repeated for all  $n$  which implies that

$$\begin{aligned} 2^n \cdot \Pr(\mathbf{a}) &= \sum_{\mathbf{m}} \sum_{\mathbf{A}} \Pr(\mathbf{A}) \cdot \delta(\mathbf{A}_{\mathbf{m}} | \mathbf{a}) \\ &= \sum_{\mathbf{m}} \Pr(\mathbf{a}) \cdot \delta(\mathbf{a}_{\mathbf{m}} | \mathbf{a}) \end{aligned}$$

which concludes the proof of Lemma 1. □

### Proof of Proposition 1.

In order to prove Proposition 1 we multiply left and right hand of Expression 1 with  $2^n$  and rewrite the resulting right hand using the definition of the parental synergy.

$$\begin{aligned}
 2^n \cdot \Pr(c) &= \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \sum_{\mathbf{A} \setminus \mathbf{A}_m} Y_{\mathbf{A} \setminus \mathbf{A}_m}(c \mid \mathbf{A}_m) \\
 &= \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \sum_{\mathbf{A} \setminus \mathbf{A}_m} \sum_{\mathbf{A}_m^*} \delta(\mathbf{A}_m \mid \mathbf{A}_m^*) \cdot \Pr(c \mid \mathbf{A}_m^*, \mathbf{A} \setminus \mathbf{A}_m) \\
 &= \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \sum_{\mathbf{A}} \delta(\mathbf{A}_m \mid \mathbf{A}) \cdot \Pr(c \mid \mathbf{A})
 \end{aligned}$$

In the second step above, an asterisk is used in order to distinguish between the two different summations over  $\mathbf{A}_m$ . In the next step,  $\mathbf{A}_m^*$  and  $\mathbf{A} \setminus \mathbf{A}_m$  are combined to  $\mathbf{A}$  after which an asterisk is not needed any more to indicate the distinction. Note that (for notational reasons),  $\delta(\mathbf{A}_m \mid \mathbf{A}_m^*)$  is changed to  $\delta(\mathbf{A}_m \mid \mathbf{A})$ , which has the same outcome. Simply rearranging terms and dividing the left and the right hand of the equation by  $2^n$  now results in the following form of Proposition 1

$$\Pr(c) = \sum_{\mathbf{A}} \Pr(c \mid \mathbf{A}) \cdot \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \delta(\mathbf{A}_m \mid \mathbf{A}) / 2^n$$

The appendix provides an example of this rewriting of Proposition 1. Since, by definition

$$\Pr(c) = \sum_{\mathbf{A}} \Pr(c \mid \mathbf{A}) \cdot \Pr(\mathbf{A})$$

and, by Lemma 1,

$$2^n \cdot \Pr(\mathbf{A}) = \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \delta(\mathbf{A}_m \mid \mathbf{A})$$

rewriting Proposition 1, together with the proof of Lemma 1, provides the proof of Proposition 1. □

**Proof of Theorem 1.**

The validity of Theorem 1 follows from the validity of Proposition 1 and its consequence Equation 2. □

## 5 Discussion

In [2], we conjectured an expression for the prior convergence error. The prior convergence error is the error which is found in the marginal prior probabilities computed for a node in a Bayesian network when the parents of this node are wrongfully assumed to be independent. The proposed expression is interesting because of its structure. The expression is composed of a part that captures the degree of dependency between the parents of the node, and of the parental synergies of the node. The parental synergies are computed from the conditional probabilities as specified for the node in a Bayesian network and act as a weighting factor, determining to what extent the degree of dependency between the parent nodes can affect the computed probabilities. The role of the parental synergy in the expression of the prior convergence error suggests that it captures a fundamental feature of a Bayesian network. In this respect, we noted that the parental synergy is related to the concepts of qualitative influence and additive synergy as defined for qualitative probabilistic networks by Wellman but is more general and more informative. In this paper we provided a proof of the correctness of the expression of the prior convergence error that we conjectured in [2].

### Acknowledgments

This research was supported by the Netherlands Organisation for Scientific Research (NWO).

### References

[1] J.H. Bolt, L.C. van der Gaag. 2004. The convergence error in loopy propagation. Paper presented at *the International Conference on Advances in Intelligent Systems: Theory and Applications*.

- [2] J.H. Bolt. 2009. Bayesian Networks: the parental synergy and the prior convergence error. In R. Serra and R. Cucchiara, editors, *the eleventh Congress of the Italian Association for Artificial Intelligence (AI\*IA'09)*, Reggio Emilia, Italy, pp 1–10, Springer-Verlag Lecture Notes in Artificial Intelligence LNAI 5883.
- [3] J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Palo Alto.
- [4] M.P. Wellman. 1990. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44:257–303.

## Appendix

In the proof of Proposition 1

$$\sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \sum_{\mathbf{A} \setminus \mathbf{A}_m} Y_{\mathbf{A} \setminus \mathbf{A}_m}(c \mid \mathbf{A}_m)$$

is rewritten in

$$\sum_{\mathbf{A}} \Pr(c \mid \mathbf{A}) \cdot \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \delta(\mathbf{A}_m \mid \mathbf{A})$$

This appendix provides an example of this conversion. Let  $\mathbf{A} = A_1, \dots, A_n, n = 2$ , then

$$\begin{aligned} & \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \sum_{\mathbf{A} \setminus \mathbf{A}_m} Y_{\mathbf{A} \setminus \mathbf{A}_m}(c \mid \mathbf{A}_m) = \\ & \sum_{A_1 A_2} \Pr(A_1 A_2) \cdot Y(c \mid A_1 A_2) + \sum_{A_1} \Pr(A_1) \cdot \sum_{A_2} Y_{A_2}(c \mid A_1) + \\ & \sum_{A_2} \Pr(A_2) \cdot \sum_{A_1} Y_{A_1}(c \mid A_2) + \sum_{A_1 A_2} Y_{A_1 A_2}(c) \end{aligned}$$

Using the definition of the parental synergy, the right hand is rewritten into

$$\begin{aligned} & \sum_{A_1 A_2} \Pr(A_1 A_2) \cdot \sum_{A_1^* A_2^*} \delta(A_1 A_2 \mid A_1^* A_2^*) \cdot \Pr(c \mid A_1^* A_2^*) + \sum_{A_1} \Pr(A_1) \cdot \sum_{A_2} \sum_{A_1^*} \delta(A_1 \mid A_1^*) \cdot \Pr(c \mid A_1^* A_2) + \\ & \sum_{A_2} \Pr(A_2) \cdot \sum_{A_1} \sum_{A_2^*} \delta(A_2 \mid A_2^*) \cdot \Pr(c \mid A_1 A_2^*) + \sum_{A_1 A_2} \delta(\cdot \mid \cdot) \cdot \Pr(c \mid A_1 A_2) \end{aligned}$$

Now an asterisk is used to distinguish between the different summations over the variables. We can rewrite the expression above into

$$\begin{aligned} & \sum_{A_1 A_2} \Pr(A_1 A_2) \cdot \sum_{\mathbf{A}} \delta(A_1 A_2 \mid \mathbf{A}) \cdot \Pr(c \mid \mathbf{A}) + \sum_{A_1} \Pr(A_1) \cdot \sum_{\mathbf{A}} \delta(A_1 \mid \mathbf{A}) \cdot \Pr(c \mid \mathbf{A}) + \\ & \sum_{A_2} \Pr(A_2) \cdot \sum_{\mathbf{A}} \delta(A_2 \mid \mathbf{A}) \cdot \Pr(c \mid \mathbf{A}) + \sum_{\mathbf{A}} \delta(\cdot \mid \cdot) \cdot \Pr(c \mid \mathbf{A}) \end{aligned}$$

Note that now the asterisk is not needed any more to distinguish between the different summations. Note furthermore that (for notational reasons)  $\delta(\mathbf{A}_m \mid \mathbf{A}_m^*)$  is changed to  $\delta(\mathbf{A}_m \mid \mathbf{A})$ , which has the same outcome. The expression above can be rewritten into

$$\sum_{\mathbf{m}=\{\{1,2\},\{1\},\{2\},\emptyset\}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \sum_{\mathbf{A}} \delta(\mathbf{A}_m \mid \mathbf{A}) \cdot \Pr(c \mid \mathbf{A})$$

which, after rearranging terms, results in

$$\sum_{\mathbf{A}} \Pr(c \mid \mathbf{A}) \cdot \sum_{\mathbf{m}} \sum_{\mathbf{A}_m} \Pr(\mathbf{A}_m) \cdot \delta(\mathbf{A}_m \mid \mathbf{A})$$

# Multi-dimensional Classification with Naive Bayesian Network Classifiers

Janneke H. Bolt

Linda C. van der Gaag

Department of Information and Computing Sciences, Utrecht University  
P.O. Box 80.089, 3508 TB, The Netherlands  
{`janneke, linda`}@uu.nl

## Abstract

Several approaches have been proposed for accommodating multi-dimensional classification with Bayesian network classifiers, among which are the use of a standard Bayesian network classifier with a compound class variable, the use of a collection of simple one-dimensional classifiers per class variable, and the use of a tailored multi-dimensional classifier. To gain fundamental insight in the differences between these types of classifier, we study their numbers of parameter probabilities, their ability to faithfully capture (in)dependencies among their variables, and their associated runtime complexity of inference.

## 1 Introduction

Bayesian network classifiers have gained considerable popularity for solving classification problems in which an instance described by a combination of feature values is to be classified in one of several distinct classes [1]. The success of naive Bayesian network classifiers especially, is readily explained from their ease of construction on the one hand and their generally good classification performance on the other hand [2]. Despite their wide applicability, Bayesian network classifiers cannot be used for solving any classification problem. Because they assume the presence of just a single class variable for example, they cannot straightforwardly model multi-dimensional classification problems. Yet, in many application domains, classification problems need to be solved in which an instance has to be assigned not to a single class but to a most likely combination of classes instead, that is, the instance is to be classified in multiple dimensions.

Over the years, researchers have proposed several different modelling approaches for multi-dimensional classification with Bayesian network classifiers, despite their basic assumption of a single class variable. One such approach is to model a multi-dimensional classification problem as a Bayesian network classifier with a single compound class variable which describes all possible combinations of class values for the problem's dimensions; we will refer to such a classifier as a model of the *Compound* type. Several researchers argued that such a *Compound* type Bayesian network classifier may easily end up with an inhibitive large number of values [3, 4]; they further argued that such a model does not properly reflect the structure of the (in)dependencies among the original class variables. Motivated by these observations, recently a new type of Bayesian network classifier was introduced which was designed specifically for multi-dimensional classification. This multi-dimensional classifier includes one or more class variables and one or more feature variables; it models the probabilistic (in)dependencies from a problem by acyclic directed graphs over the class variables and over the feature variables separately, and further connects the two sets of variables by a bipartite directed graph [3, 7]. The fully naive multi-dimensional classifier more specifically, with empty class and feature graphs, shares the advantage of ease of construction with the one-dimensional naive Bayesian network classifier. For ease of reference, we will call a multi-dimensional Bayesian network classifier, a model of the *Multi* type. A third approach to multi-dimensional classification is to construct a collection of simple one-dimensional Bayesian network classifiers, each of which pertains to a single class variable; we say that such a collection of classifiers is a model of the *Split* type. As pointed out by several researchers, *Split* type Bayesian network classifiers cannot capture any interactions among the dimensions defined by

their class variables [3, 4]. The classifier may moreover return a combination of class values which is not the most likely explanation of the observed combination of feature values.

For a specific multi-dimensional classification problem, in essence any of the three types of naive Bayesian network classifier can be developed. For choosing which type to actually use however, little fundamental insight is available by which the three types of classifier can be compared. Intuition suggests for example, that Bayesian network classifiers of the *Split* type will be inferior to classifiers of the *Compound* and *Multi* types, as a consequence of their strong independency assumptions. Preliminary experimental results further suggest better performance of classifiers of the *Multi* type over those with compound class variables, as a result of the smaller number of parameter probabilities to be estimated from the available data [3]. As these intuitions and findings require a more formal underpinning, we initiated a study into the fundamental properties of the three types of Bayesian network classifier in view of multi-dimensional classification. In our study, we focused so far on complete, naive models because of their evident popularity, that is, we focused on Bayesian network classifiers which include all available feature variables yet do not include any dependencies that may exist among them given the various class variables. In this paper, we compare the numbers of parameter probabilities of the three types of classifier and the induced runtime complexity of inference; we further compare the three types of Bayesian network classifier in terms of their ability to faithfully represent the independencies from a true joint probability distribution.

The paper is organised as follows. In Section 2, we provide some preliminaries on Bayesian network classifiers. In Section 3, we present our running example and construct the three types of naive Bayesian network classifier to describe the classification problem from the example. In Section 4 we compare various properties of the three types of classifier. The paper ends with our concluding observations in Section 5.

## 2 Preliminaries

We briefly review some concepts from Bayesian networks in general, and introduce one-dimensional naive Bayesian network classifiers and fully naive multi-dimensional network classifiers more specifically.

A Bayesian network is a probabilistic graphical model which describes a joint probability distribution  $\Pr$  over a set of discrete random variables  $\mathbf{V}$  [5, 6]. The variables and their interrelationships are modelled in an acyclic directed graph. Each node in this graph captures a variable  $V \in \mathbf{V}$ , which can adopt a value from among an associated domain of values  $\Omega(V)$ . The arcs of the graph model the probabilistic (in)dependencies between the variables through the well-known d-separation criterion [5]. We say that two variables  $V_i$  and  $V_j$  are d-separated in the graph by the set of variables  $\mathbf{X}$ , written  $\langle V_i, V_j \mid \mathbf{X} \rangle$ , if every chain between  $V_i$  and  $V_j$  contains either a variable from  $\mathbf{X}$  with at least one emanating arc, or a variable  $V_k$  with two incoming arcs such that neither  $V_k$  itself nor any of its descendants are included in  $\mathbf{X}$ ;  $V_i$  and  $V_j$  are then considered mutually independent given  $\mathbf{X}$ , which will be denoted as  $I(V_i, V_j \mid \mathbf{X})$ . Associated with each variable  $V$  in the network's graph moreover, is a set of parameter probabilities  $\Pr(V \mid \pi(V))$  which describe the probabilities over  $V$  given all possible values for its parents  $\pi(V)$ . For computing prior and posterior probabilities over the separate variables of a network, efficient algorithms are available [6].

A naive Bayesian network classifier is a Bayesian network of restricted topology, tailored to modelling and solving classification problems. It includes a single class variable  $C$  and one or more feature variables  $F_i$ . Each feature variable has a single incoming arc, from the class variable. The feature variables thus are modelled as being mutually independent given the class variable. A naive Bayesian network classifier is used for computing the posterior probability distribution  $\Pr(C \mid \mathbf{f})$  over the class variable, given a joint value combination  $\mathbf{f}$  for the set  $\mathbf{F}$  of feature variables. From the computed distribution, the most likely value of the class variable is established (breaking ties randomly), and returned as the classifier's output.

Fully naive multi-dimensional Bayesian network classifiers are also Bayesian networks of restricted topology, but unlike naive Bayesian network classifiers they are tailored to classification in multiple dimensions. A fully naive multi-dimensional classifier includes one or more class variables  $C_j$  and multiple feature variables  $F_i$  [3, 7]. The class variables do not have any arcs between them, and hence are modelled as being mutually independent. Each feature variable has incoming arcs from all class variables; the feature variables are unconnected otherwise, and thus mutually independent given the class variables. A multi-dimensional Bayesian network classifier is used for computing the joint probability distribution  $\Pr(\mathbf{C} \mid \mathbf{f})$  over the set of class variables, given a joint combination of values  $\mathbf{f}$  for the set of feature variables. From the computed posterior distribution, the most likely joint value combination for the set of class variables is established (again breaking ties at random) and returned as the classifier's output.

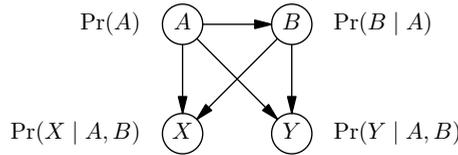


Figure 1: The (in)dependency graph of the joint probability distribution  $\Pr$  of the running example

For multi-dimensional classification in general, it does not suffice to consider each class variable separately, as the most likely joint value combination of these variables not necessarily equals the combination of their most likely values. As an example, we consider the Bayesian network with the graph from Figure 1 and the parameters probabilities from Table 1. We suppose that the combination of feature values  $X = x, Y = y$  is to be classified. From the network, we compute the joint distribution over the class variables to be  $\Pr(a, b | x, y) = 0.009$ ,  $\Pr(a, \bar{b} | x, y) = 0.391$ ,  $\Pr(\bar{a}, b | x, y) = 0.365$  and  $\Pr(\bar{a}, \bar{b} | x, y) = 0.235$ . The most likely joint value combination for  $A$  and  $B$  thus is  $a, \bar{b}$ . From  $\Pr(a | x, y) = 0.4$  and  $\Pr(b | x, y) = 0.374$  however, their most likely values are found to be  $\bar{a}$  and  $\bar{b}$ .

### 3 A Running Example

For a running example throughout the paper, we consider a simple multi-dimensional classification problem with the two class variables  $A$  and  $B$ , and the two feature variables  $X$  and  $Y$ . We assume a joint probability distribution  $\Pr$  over the four variables, which embeds the (in)dependencies described by the graph from Figure 1. We further assume that joint combinations of values for the two feature variables have to be classified in the dimensions defined by the two class variables. For a given joint value combination  $\mathbf{f}$  for the variables  $X$  and  $Y$  therefore, we are interested in the probability distribution  $\Pr(A, B | \mathbf{f})$ , and in the most likely joint value combination for the class variables  $A$  and  $B$  more specifically. Since in this paper, we would like to focus on the differences between the various approaches to multi-dimensional classification in the absence of direct dependencies between the feature variables involved, we have chosen for our running example a distribution that embeds this property of independency.

From the example distribution  $\Pr$ , we now construct the three types of naive Bayesian network classifier under study for multi-dimensional classification. The first constructed classifier is a classifier of the *Compound* type. More specifically, it is a one-dimensional naive Bayesian network classifier with the compound class variable  $AB$ . This class variable  $AB$  takes for its domain the full Cartesian product of the domains of the two original class variables  $A$  and  $B$ , that is,  $\Omega(AB) = \Omega(A) \times \Omega(B)$ . The parameter probabilities for the variable  $AB$  are computed from the original distribution to be equal to  $\Pr(AB) = \Pr(B | A) \cdot \Pr(A)$ . The two feature variables  $X$  and  $Y$  are connected directly to the compound class variable, as in naive Bayesian network classifiers in general. These variables inherit their parameter probabilities directly from the original representation of  $\Pr$ , with  $\Pr(X | AB) = \Pr(X | A, B)$  and  $\Pr(Y | AB) = \Pr(Y | A, B)$ . The graph of the resulting *Compound*-type classifier is depicted in Figure 2. From this classifier, posterior probability distributions  $\Pr(AB | \mathbf{f})$  are computed, from which the most likely value of the compound class variable  $AB$  given  $\mathbf{f}$  is established.

The second classifier constructed for the example problem is a fully naive multi-dimensional Bayesian network classifier. This classifier of the *Multi* type includes the four variables directly, without requiring any transformation. The two class variables are not connected in the classifier; the two feature variables have incoming arcs from both class variables, and are unconnected otherwise. The variables  $A, X$  and  $Y$  inherit their parameter probabilities from the original model. The parameters for the class variable  $B$  are computed, by conditioning and marginalisation, to be equal to  $\Pr(B) = \sum_A \Pr(B | A) \cdot \Pr(A)$ . The graph of the resulting *Multi* classifier is shown in Figure 3. A posterior joint probability distribution over the class variables is now computed as  $\Pr(A, B | \mathbf{f}) = \Pr(A | \mathbf{f}) \cdot \Pr(B | A, \mathbf{f})$ . From this joint distribution, the

Table 1: A set of parameter probabilities for the graph of the running example

$\Pr(x   ab) = 0.1$	$\Pr(y   ab) = 0.2$	$\Pr(a) = 0.6$
$\Pr(x   a\bar{b}) = 0.7$	$\Pr(y   a\bar{b}) = 0.5$	$\Pr(b   a) = 0.1$
$\Pr(x   \bar{a}b) = 0.2$	$\Pr(y   \bar{a}b) = 0.2$	$\Pr(b   \bar{a}) = 0.9$
$\Pr(x   \bar{a}\bar{b}) = 0.9$	$\Pr(y   \bar{a}\bar{b}) = 0.9$	

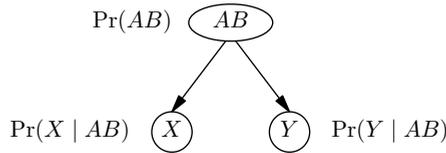


Figure 2: The graph of the *Compound*-type classifier for the distribution  $\Pr$  of the running example

most likely joint value combination for the two class variables is established.

The third constructed classifier is a *Split*-type classifier, which is composed of two separate components. One of the components includes the class variable  $A$  and allows the classification of feature combinations  $\mathbf{f}$  in the dimension defined by  $A$ ; the other component includes the class variable  $B$ . Both components of the *Split*-type classifier include references to the original feature variables  $X$  and  $Y$ ; to distinguish between these references, we annotate them by the class variable from the component they belong to. The graph of the *Split*-type classifier now is shown in Figure 4; we note that the two components can basically be looked upon as two separate one-dimensional naive Bayesian network classifiers. The class variable  $A$  in the overall classifier inherits its parameter probabilities directly from the representation of the original distribution  $\Pr$ ; the parameter probabilities  $\Pr(B)$  are computed to be  $\Pr(B) = \sum_A \Pr(B | A) \cdot \Pr(A)$  by conditioning and marginalisation. The parameter probabilities  $\Pr(X_A | A)$  for the reference variable  $X_A$  in the component for  $A$ , are calculated as  $\Pr(X_A | A) = \sum_B \Pr(X | A, B) \cdot \Pr(B | A)$  from the original representation of  $\Pr$ ; the parameters  $\Pr(Y_A | A)$  and the parameter probabilities for the reference variables from the other component are computed analogously. From the *Split*-type classifier, the most likely joint value combination for the class variables  $A$  and  $B$  is computed as the combination of the most likely values for  $A$  and  $B$  separately; these values are established from the posterior probability distributions  $\Pr(A | \mathbf{f})$  and  $\Pr(B | \mathbf{f})$  computed from the two components of the overall classifier, respectively. We note that the returned joint value combination for the class variables  $A$  and  $B$  equals the most likely combination of values established from  $\Pr(A, B | \mathbf{f}) = \Pr(A | \mathbf{f}) \cdot \Pr(B | \mathbf{f})$ .

## 4 Properties of Naive Classifiers for Multi-dimensional Classification

The three types of naive Bayesian classifier for multi-dimensional classification are now compared in terms of various fundamental properties. Their numbers of parameter probabilities are detailed in Section 4.1. In Section 4.2 we investigate the ability of the classifiers to capture the (in)dependencies from a problem at hand. The runtime complexity of inference with the classification models is studied in Section 4.3.

### 4.1 The numbers of parameter probabilities required

Because naive Bayesian classifiers are of fixed topology, their construction from data amounts to just learning all required parameter probabilities. The number of probabilities to be estimated from the data constitutes an important property of such a classifier, as it is related directly to the amount of bias in the estimates obtained. We now study the numbers of parameter probabilities of our three types of naive classifier.

We begin by considering the numbers of parameter probabilities of the three classifiers from the Figures 2, 3 and 4. For example, let  $k_A = 2$  and  $k_B = 3$  be the numbers of possible values for the class variables  $A$  and  $B$  respectively, and let  $\ell_X = 4$  and  $\ell_Y = 5$  be the sizes of the value domains of  $X$  and  $Y$ . The classifier of the *Compound* type, from Figure 2, includes for its compound class variable  $AB$  prior probabilities for all value combinations for the two original class variables  $A$  and  $B$ ; for the compound variable therefore,  $k_A \cdot k_B = 6$  probabilities are specified. For each of the feature variables, the classifier includes conditional

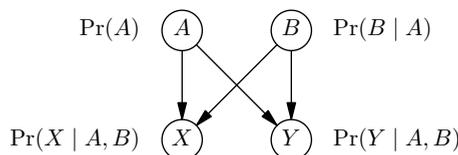


Figure 3: The graph of the *Multi*-type classifier for the distribution  $\Pr$  of the running example

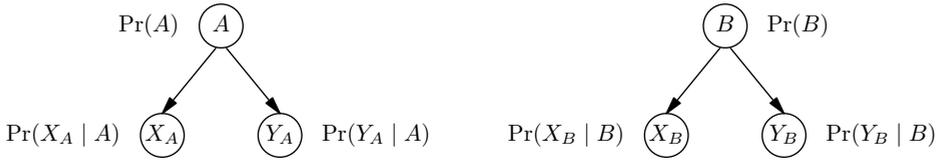


Figure 4: The graph of the *Split*-type classifier for the distribution  $\text{Pr}$  of the running example

probability distributions for each value of the compound class variable separately. For the variable  $X$  therefore,  $\ell_X \cdot k_A \cdot k_B = 24$  probabilities are included; similarly,  $\ell_Y \cdot k_A \cdot k_B = 30$  conditional probabilities are specified for the variable  $Y$ . The total number of parameter probabilities of the *Compound*-type classifier thus is  $(\ell_X + \ell_Y + 1) \cdot k_A \cdot k_B = 60$ . We now address the number of parameter probabilities of the *Multi* type classifier from Figure 3. This classifier includes  $k_A + k_B = 5$  prior probabilities for its two class variables. For the two feature variables, it specifies  $(\ell_X + \ell_Y) \cdot k_A \cdot k_B = 54$  estimates. The total number of probabilities of the *Multi* type classifier thus equals 59. We observe that the numbers of probabilities of the *Compound*- and *Multi*-type classifiers do not differ much, and in fact are both just slightly smaller than the 60 parameter probabilities of the original representation of the distribution  $\text{Pr}$ . The *Split*-type classifier from Figure 4 to conclude, includes fewer parameter probabilities. The one-dimensional classifier for the class variable  $A$  specifies  $(\ell_X + \ell_Y + 1) \cdot k_A = 20$  parameter probabilities; the classifier for  $B$  has  $(\ell_X + \ell_Y + 1) \cdot k_B = 30$  parameters. The total number of parameter probabilities of this classifier thus equals 50.

Table 2 summarises the numbers of parameter probabilities of the three types of naive Bayesian network classifier more in general, assuming  $n$  class variables with  $k$  values each and  $m$  feature variables with  $\ell$  values each. By comparing their numbers of associated parameter probabilities, we conclude that a naive Bayesian network classifier of the *Split* type requires the smallest number of parameters. While the classifiers of the *Compound* and *Multi* types include numbers of parameter probabilities that are exponential in the number of class variables  $k$ , the *Split*-type classifier has a polynomial number of parameters in  $k$ . We further note that the naive classifier of the *Compound* type requires the largest number of parameters. Yet, this type of classifier requires just  $k^n - n \cdot k$  more parameters than classifiers of the *Multi* type. For classification problems involving a relatively large number of class variables with small numbers of values, the difference in the numbers of parameters of the two types of classifier is almost negligible, as for both types the number of parameter probabilities is dominated by the numbers of parameters required for the feature variables. In fact, adding a tree-like class graph over the class variables of a multi-dimensional Bayesian network classifier would not add much to the complexity of the model.

## 4.2 The ability of representation

The graph of a Bayesian network in general describes the probabilistic (in)dependencies among its variables through the d-separation criterion: if two variables are d-separated in the graph, they are considered independent in the represented probability distribution. Since naive Bayesian network classifiers have a fixed topology of their graphs, they assume specific independencies to hold. If the true independencies of a joint probability distribution do not match those of a naive classifier, then this classifier cannot faithfully capture the true distribution. As a consequence, any probabilities computed from the classifier may then be inaccurate. Since their ability of representation is related directly to classification accuracy, we study this property in our three types of naive Bayesian network classifier.

We begin by establishing the independencies of the original probability distribution  $\text{Pr}$ , from its graph in Figure 1. Building upon the observation that the only arc missing from the graph is an arc between the two feature variables, we find that  $\langle X, Y \mid \{A, B\} \rangle$  defines the only represented independency: the

Table 2: The numbers of parameter probabilities and the runtime complexity of inference, for the three types of naive classifier, assuming  $n$  class variables with  $k$  values each and  $m$  feature variables with  $\ell$  values each

Classifier type	Number of parameters	Runtime complexity
<i>Compound</i>	$(m \cdot \ell + 1) \cdot k^n$	$\mathcal{O}(m \cdot \ell \cdot k^n)$
<i>Multi</i>	$m \cdot \ell \cdot k^n + n \cdot k$	$\mathcal{O}(m \cdot \ell \cdot k^n)$
<i>Split</i>	$(m \cdot \ell + 1) \cdot n \cdot k$	$\mathcal{O}(m \cdot n \cdot \ell \cdot k)$

feature variables  $X$  and  $Y$  are independent given the class variables  $A$  and  $B$ . The two feature variables may be dependent a priori however, and may remain to be so given each class variable separately. The two class variables cannot be assumed to be independent. We now compare these independencies against the independencies that are assumed to hold by the three types of naive classifier. From the graph of the classifier of the *Compound* type, from Figure 2, we observe that  $\langle X, Y \mid \{AB\} \rangle$ . From this d-separation finding, we have that the feature variables  $X$  and  $Y$  are assumed to be independent given the compound class variable  $AB$ . Since this compound variable models the joint behaviour of the original class variables  $A$  and  $B$ , we conclude that the *Compound*-type classifier captures the same independencies as the original distribution. For the classifier of the *Multi* type, we find from Figure 3 that it equally assumes that the independency  $I(X, Y \mid \{A, B\})$  holds for the two feature variables. In addition, we find from the graph that  $\langle A, B \mid \emptyset \rangle$ . The classifier of the *Multi* type thus makes the additional assumption  $I(A, B \mid \emptyset)$  of independency of the two class variables.

For the classifier of the *Split* type, we find that the two class variables  $A$  and  $B$  are a priori independent, since we read  $\langle A, B \mid \emptyset \rangle$  from the graph from Figure 4. We further find that the two class variables remain independent given each or both (references to the) feature variables. We therefore conclude that the *Split* type classifier assumes that the independencies  $I(A, B \mid \mathbf{G})$ ,  $\mathbf{G} \subseteq \{X_A, Y_A, X_B, Y_B\}$ , hold for its class variables. For the (references to the) feature variables, we further find that for example the independencies  $I(X_A, Y_A \mid \{A\} \cup \mathbf{G})$ ,  $\mathbf{G} \subseteq \{B, X_B, Y_B\}$ , and  $I(X_B, Y_B \mid \{B\} \cup \mathbf{G})$ ,  $\mathbf{G} \subseteq \{A, X_A, Y_A\}$ , are assumed to hold. In addition, each class variable is assumed to be independent from the feature variables that are not included in the same component. We would like to note that although the two one-dimensional models from the overall *Split*-type classifier are separate components in Figure 4, they are not independent in view of multi-dimensional classification. Since the feature variables in the two components basically are references to the same underlying variables  $X$  and  $Y$ , in using the *Split*-type classifier for solving multi-dimensional classification problems the references to  $X$  and  $Y$  are forced to assume the same values.

When performing classification in multiple dimensions, the most likely combination of class values  $\mathbf{c}$  is sought for an instance described by a combination of feature values  $\mathbf{f}$ . The classification thus amounts to establishing  $\operatorname{argmax}_{\mathbf{c}} \{\Pr(\mathbf{c} \mid \mathbf{f})\}$  from the represented distribution. As argued above, the three types of classifier under study capture different independencies which are exploited for computing the required posterior probability distribution. We now study the computations used by the different classifier types upon performing the classification task. We begin by noting that essentially the following probabilities need to be computed for all  $\mathbf{c} \in \Omega(\mathbf{C})$ ,  $\mathbf{c} = c_1, \dots, c_n$ :

$$\Pr(\mathbf{c} \mid \mathbf{f}) = \frac{\Pr(\mathbf{f} \mid c_1, \dots, c_n) \cdot \Pr(c_1, \dots, c_n)}{\sum_{C_1, \dots, C_n} \Pr(\mathbf{f} \mid C_1, \dots, C_n) \cdot \Pr(C_1, \dots, C_n)}$$

The *Compound*-type classifier computes:

$$\Pr_{\mathbf{C}}(\mathbf{c} \mid \mathbf{f}) = \frac{\prod_{i=1}^m \Pr(f_i \mid c_1 \dots c_n) \cdot \Pr(c_1 \dots c_n)}{\sum_{C_1 \dots C_n} \prod_{i=1}^m \Pr(f_i \mid C_1 \dots C_n) \cdot \Pr(C_1 \dots C_n)}$$

where  $C_1 \dots C_n$  denotes the constructed compound class variable. We note that this type of classifier assumes the independencies  $I(F_i, F_j \mid \mathbf{C})$  to hold for any  $F_i, F_j \in \mathbf{F}, i \neq j$ . Because of this assumption, the term  $\Pr(\mathbf{f} \mid C_1, \dots, C_n)$  is equal to  $\prod_{i=1}^m \Pr(f_i \mid C_1 \dots C_n)$ , and hence is replaced as shown above. In the *Multi*-type classifier, the probabilities  $\Pr(\mathbf{c} \mid \mathbf{f})$  are computed as

$$\Pr_M(\mathbf{c} \mid \mathbf{f}) = \frac{\prod_{i=1}^m \Pr(f_i \mid c_1, \dots, c_n) \cdot \prod_{j=1}^n \Pr(c_j)}{\sum_{C_1, \dots, C_n} \prod_{i=1}^m \Pr(f_i \mid C_1, \dots, C_n) \cdot \prod_{j=1}^n \Pr(C_j)}$$

We note that this type of classifier assumes the independencies  $I(C_i, C_j \mid \emptyset)$ ,  $C_i, C_j \in \mathbf{C}, i \neq j$ , to hold among its class variables, in addition to the conditional independencies  $I(F_i, F_j \mid \mathbf{C})$  of the feature variables. Because of the assumption of mutual independency of the class variables, the term  $\Pr(C_1 \dots C_n)$  is equal to the term  $\prod_{j=1}^n \Pr(C_j)$ , and thus is replaced as shown in the formula above. In a *Split*-type classifier to conclude, the probabilities  $\Pr(\mathbf{c} \mid \mathbf{f})$  are computed as

$$\Pr_S(\mathbf{c} \mid \mathbf{f}) = \frac{\prod_{i=1}^m \prod_{j=1}^n \Pr(f_i \mid c_j) \cdot \Pr(c_j)}{\sum_{C_1, \dots, C_n} \prod_{i=1}^m \prod_{j=1}^n \Pr(f_i \mid C_j) \cdot \Pr(C_j)}$$

The strong independency properties holding in this type of classifier result in the formula above, in which the additional replacement of  $\Pr(f_i \mid C_1, \dots, C_n)$  by  $\prod_{j=1}^n \Pr(f_i \mid C_j)$  is observed.

In conclusion, we would like to note that from the *Compound*- via the *Multi*- to the *Split*-type classifier for multi-dimensional classification, more and more independency properties are assumed, each of which may introduce inaccuracies in the computed probabilities from which the most likely combination of class values is established. The inaccuracies in the computed joint distribution  $\Pr(\mathbf{C} \mid \mathbf{f})$  over the class variables may in fact result in a different classification of an instance. As an example, we consider again the probability distribution  $\Pr$  represented by the graph from Figure 1, with the parameters from Table 1. In the initial model an instance  $x, y$  is classified as  $a, \bar{b}$ . In the *Compound* model, this instance is necessarily classified as  $a, \bar{b}$  as well since the compound classifier represents an equivalent probability distribution. In the *Multi* model, however, the assumed independency between  $A$  and  $B$  results in the deviating classification  $\bar{a}, \bar{b}$ . In the *Split* model, further simplification results in the classification of this instance in  $a, b$ . Since the *Split* model is more simplified than the *Multi* model, one would expect the *Multi* model to perform better. Our simple example shows however, that inaccuracies may have counterbalancing effects, which can result in an unexpected better performance of the *Split*-type classifier.

### 4.3 Complexity of inference

The most efficient algorithm for computing probability distributions from a Bayesian network is the junction-tree propagation algorithm. This algorithm builds upon an auxiliary graphical structure which is constructed from the digraph of the Bayesian network in three steps. In the first step, the parents of each variables having two or more incoming arcs are mutually connected, after which the directions of the all arcs are dropped. In the second step, the resulting undirected graph is triangulated by adding appropriate undirected edges, that is, edges are inserted until the graph does no longer include cycles of length four or more without a shortcut. From the triangulated graph, the maximum cliques are determined, which subsequently are organised in a specific tree structure. The domains of the cliques equal the Cartesian products of the included variables. The domain sizes determine the complexity of the inference.

We consider again the example joint probability distribution  $\Pr$ , and the three types of naive Bayesian network classifier for multi-dimensional classification from Figures 2, 3 and 4. For the one-dimensional naive Bayesian network classifier of the *Compound* type, the propagation will use a junction tree with two cliques. One of the cliques includes the compound class variable and the feature variable  $X$ , and the other clique includes the feature variable  $Y$  also with the compound class variable  $AB$ . The domain sizes of the two cliques are  $\ell_X \cdot k_A \cdot k_B$  and  $\ell_Y \cdot k_A \cdot k_B$ , which makes the junction-tree propagation algorithm run in  $\mathcal{O}((\ell_X + \ell_Y) \cdot k_A \cdot k_B)$  time. The junction tree constructed for the multi-dimensional model, will again include two cliques, this time of three variables each. The one clique includes the two class variables and the feature variable  $X$ , and the other one has the feature variable  $Y$  in addition to the two class variables. The domain sizes of the two cliques again are  $\ell_X \cdot k_A \cdot k_B$  and  $\ell_Y \cdot k_A \cdot k_B$  as for the classifier of the *Compound* type, which makes junction-tree propagation run in  $\mathcal{O}((\ell_X + \ell_Y) \cdot k_A \cdot k_B)$  time on the *Multi* model. For the two naive Bayesian network classifiers constituting the *Split* model, a junction tree with four cliques is constructed: these cliques each include a single class variable and a single feature variable, and have domain sizes  $\ell_X \cdot k_A$ ,  $\ell_Y \cdot k_A$ ,  $\ell_X \cdot k_B$  and  $\ell_Y \cdot k_B$  respectively. Junction-tree propagation will run in  $\mathcal{O}((k_A + k_B) \cdot (\ell_X + \ell_Y))$  time. More in general, Table 2 summarises the runtime properties of the junction-tree propagation algorithm for naive Bayesian network classifiers which include  $n$  class variables with  $k$  values each and  $m$  feature variables with  $\ell$  values each.

From the above considerations, we conclude that probabilistic inference is the most efficient in naive classifiers of the *Split* type: while the runtime properties of the junction-tree propagation algorithm include an exponential term for the *Compound* and *Multi* classifiers, the algorithm will run in bi-linear time in the numbers of class and feature variables involved. Only for multi-dimensional problems with a relatively small number of class variables with small domain sizes can the runtime complexity of the *Compound* and *Multi* models approximate the runtime complexity of the classifier of the *Split* type. We observe that the *Compound* and *Multi* model induce the same runtime properties of the junction-tree propagation algorithm, as for both types of classifier the same auxiliary structure is derived.

## 5 Discussion

In many application domains, classification problems need to be solved in which an instance has to be assigned to a most likely combination of classes. For accommodating such multi-dimensional classification in Bayesian network classifiers, different types of model can be constructed. In a classifier of the *Compound*

type a single compound class variable is constructed which describes all possible combinations of class values for the dimensions of the problem; in a classifier of the *Multi* type, multiple class variables are included and in a classifier of the *Split* type, a collection of simple Bayesian network classifiers is used, one for each class variable. Little insight is available as yet for comparing the different types of classifier. In this paper, we initiated a study into the fundamental properties of the three types of Bayesian network classifier for multi-dimensional classification. Upon doing so, we focused on complete, naive classifiers. We compared the different classifier types with respect to their number of parameter probabilities, their ability of representing the (in)dependencies of probability distributions, and the complexity of inference they induce.

Although Bayesian network classifiers of the *Multi* type are being propagated as good alternatives for *Compound*-type classifiers, we argued that the differences in their numbers of parameter probabilities and in the complexity of inference are quite small. Since at the same time their accuracy of representation may be reduced, we feel that we cannot expect *Multi*-type classifiers to outperform classifiers of the *Compound* type, if complete, naive models as assumed. While classifiers of the *Split* type are clearly inferior in their representation of the probability distribution of a problem domain, they require substantially fewer parameter probabilities for their specification, which serves to reduce their bias from available data. Inference in these models moreover, is very efficient. In the near future, we hope to gain more insight in the conditions under which *Split*-type classifiers can be expected to show satisfactory performance.

In this paper, we studied complete, naive Bayesian network classifiers only. Different results are expected, for example, for classifiers which allow interactions between their class variables and/or feature variables. Classifiers built using feature selection may, for example, amplify the differences between classifiers of the *Compound* and *Multi* types with respect to their numbers of parameters and the complexity of inference. When feature selection would result in relatively small sets of class parent for the feature variables, the number of parameter probabilities involved would begin to diverge between the *Multi*- and *Compound*-type classifiers in favour of the former type. Also the domain sizes of the cliques in the junction tree would then become smaller in the *Multi* model, resulting in more efficient inference. Our further research efforts will be directed at increased fundamental insight in more complex types of Bayesian network classifier for multi-dimensional classification in the effects of different learning paradigms on the formal properties of the different types of classifier.

**Acknowledgement.** This research was funded by the Netherlands Organisation for Scientific Research.

## References

- [1] N. Friedman, D. Geige, M. Goldszmidt (1997). Bayesian network classifiers. *Machine Learning*, vol. 29, pp. 131 – 164.
- [2] D.J. Hand, K. You (2001). Idiot’s Bayes – not so stupid after all ? *International Statistical Review*, vol. 69, pp. 385 – 398.
- [3] L.C. van der Gaag, P.R. de Waal (2006). Multi-dimensional Bayesian network classifiers. In: M. Studený, J. Vomlel (eds). *Proceedings of the Third European Workshop on Probabilistic Graphical Models*, pp. 107 – 114.
- [4] J.D. Rodriguez, J.A. Lozano (2008). Multi-objective learning of multi-dimensional Bayesian classifiers. *Eighth International Conference on Hybrid Intelligent Systems*, pp. 501 – 506.
- [5] J. Pearl (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.
- [6] F.V. Jensen, T.D. Nielsen (2007). *Bayesian Networks and Decision Graphs*, 2nd ed., Springer Verlag.
- [7] P.R. de Waal, L.C. van der Gaag (2007). Inference and learning in multi-dimensional Bayesian network classifiers. In: K. Mellouli (ed). *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, LNCS 4724, Springer, Berlin, pp. 501 – 511.

# Sharing Blood: A decentralised trust and sharing ecosystem based on the Vampire Bat

Bulten, W. <sup>a</sup>      Haselager, W.F.G. <sup>b</sup>      Sprinkhuizen-Kuyper, I.G. <sup>b</sup>

<sup>a</sup> *Department of Artificial Intelligence, Radboud University Nijmegen, The Netherlands*

<sup>b</sup> *Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour,  
The Netherlands*

## Abstract

Vampire bats manage to live longer by trusting their fellow roost mates and getting donated to in return. We have modelled this interaction by creating a biological plausible decentralised trust and sharing system. In a simulated 3D environment the performance has been tested by groups of artificial bats showing a significant increase in life span as a result of the bat trust ecosystem. To further test the system, groups of cheaters were added to influence the population of trusters. Even though cheaters have a negative influence on the population of trusters we have found that this is (for the most part) not the result of their cheating behaviour. In other words, trust pays and is robust.

## 1 Introduction

Trust is intertwined with our whole life. We trust our family and friends to support us, doctors to take care of us and governments to protect us. Not only in our lives, but also in the lives of other animals, trust plays a significant part. Numerous cases have been reported of animals trusting and cooperating [1]. Related to this is what Trivers [4] calls *reciprocal altruism*: behaviour that benefits a not closely related organism while being disadvantageous for the donor. In this paper we will focus on the vampire bat and use computational modelling of altruistic food sharing.

Vampire bats live on a diet of blood, need a lot of it to fulfil their nutritional needs and can easily consume 33% of their own weight in blood [10]. Importantly, research on the common vampire bat (the *desmodus rotundus*) showed that bats younger than two years have a 30% chance to fail in finding a sufficient meal. For mature bats this chance to fail is still 7%. Also, within a colony, not all bats fail at the same time in finding food [5, 9].

Vampire bats can only survive for no more than 48 to 60 hours without food. Based on the probability of finding an appropriate meal, it seems unlikely that these bats can reach the observed ages of up to eighteen years (especially for females). Wilkinson noticed from these numbers that the annual mortality, based on the probabilities, should be 82% whereas the observed mortality with real bats is only 24%. How does this happen?

When a bat has sufficient blood available it regularly, when certain criteria are met, donates blood to less fortunate roost mates. Though this lowers the fitness of the donor it even occurs between non-relatives in a colony. Wilkinson defined five criteria that have to be met before behaviour can qualify as reciprocal altruism (instead of for example kin selection) [8], the vampire bat matches all of these:

**(1) The behaviour must reduce a donor's fitness relative to the selfish alternative:** A donor bat regurgitates about 5 millilitres of blood and with that action loses approximate six hours of life time. When it would choose for the selfish alternative it will have more time to find a new food source. **(2) The fitness of the recipient must be elevated relative to the non-recipient:** The recipient gains about 18 hours due to the donation and therefore benefits more than the donor. This is a result of a non-linear relation between body weight and time. The more food an agent has, the higher its decrease in body weight per hour. **(3)**

**Performance of the behaviour must not depend on receipt of an immediate benefit:** The donor bat cannot receive blood from the recipient on the spot because the recipient has not enough food to share.

**(4) A mechanism for detecting individuals who receive but never pay altruistic costs has to exist:** It has been suggested that vampire bats use grooming to detect cheaters [7]. During grooming bats would inspect stomach sizes of other bats and use that to keep track of feeding records. **(5) A large number of opportunities to exchange aid must exist within each individual's lifetime:** The social structure of female vampire bats is fairly stable [6]. Female offspring stay in their natal group and usually only move when their mother moves or dies.

Wilkinson used simulations to calculate the benefit of altruistic acts. His Monte Carlo simulation used fixed association values between bats, fixed chances of finding food (always 0.9) and only 11 bats. It is our aim in this paper to, first, improve his simulation and, second, extend it by introducing a new group of agents: cheaters. A first block of simulations will be used to answer our first main research question: What does the trust system contribute in terms of bat performance, i.e. to what extent is their lifetime prolonged, compared to a control group, and how, specifically, is this contribution influenced by the availability of food?

The second block of simulations focusses on cheaters and will try to answer our second group of research questions: Do cheaters shorten the life span of trusting agents, is this caused by their cheating behaviour or just by the presence of a non-sharing group and how does the size of the cheater population influence this effect?

## 2 Methods

Two blocks of simulations were run where the first block of simulations test the performance of the system and tries to be biological plausible; the second block focuses on the influence of cheaters.

### 2.1 Environment

The system will be modelled and tested in a virtual environment that resembles a simplified version of the biome vampire bats live in. The bats are modelled using the *Python* language and the open-source software *Breve* [2] which has also been used to run the simulations. Some inspiration has been taken from existing *Breve* simulations (including [3]) though their influence was small as these were often written in the *Breve* programming language *Steve*.

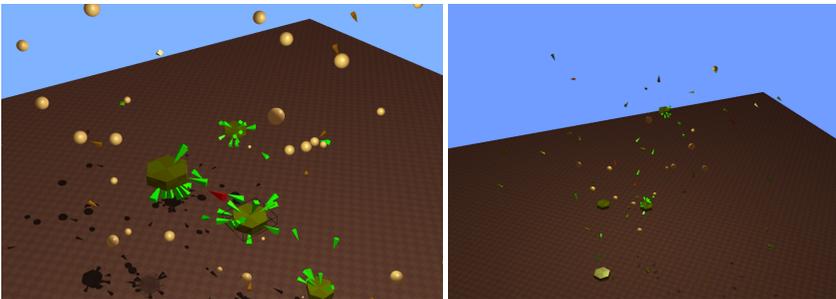


Figure 1: Screenshots of the setup showing food sources (balls), agents (polygons) and nests (polygon disks). An agent's colour resembles the current energy level and ranges from green (saturated) to red (starving).

The environment consisted of a 3D world with a fixed size and agents were only able to navigate within that area. Each simulation used the same 3D world, an example can be seen in Figure 1. Before the simulation started the world was filled with four nests, a variable amount of food sources and agents. Agents return to the nest after feeding or when the night ends. Before each hunting period food sources were reset (filled with blood) and placed at random locations.

### 2.2 Agents

All agents in the simulations share a set of common behaviours, these include exploring, locating food, feeding, returning and leaving the nest and dying. These common or *base* behaviours are implemented in a

*base agent* which has been used as an abstract class for the other agents. By further implementing this base agent a total of three different types of agents were created to live in the environment:

**Control Agents** only contain the behaviours included in the base agent. They will try to find food but will never share or beg. **Trust Agents** resemble real vampire bats in the sense that they have the ability to trust another agent and are able to donate and receive food from other agents. **Cheating Agents** will never donate food to another agent. Instead, cheating agents will try to receive as much food possible from other agents.

### 2.2.1 Basic Behaviour

The most important influence in the basic behaviour of the agents is the day-night cycle. When the night falls agents will leave their nest and start searching for food and will, whether they have found food or not, return to the nest before sunrise.

When wandering through the world an agent can detect a food source and fly to it when it is in the line of sight of the agent<sup>1</sup>. When a food source is detected and it is not yet occupied by another bat, an agent will fly to the source until it is within feeding distance and will feed until it is fully saturated or the source is depleted.

Besides returning to the nest at dawn, there are two more situations in which an agent returns to the nest. First of all, if an agent is fully saturated, because of a successful hunt, it will stop exploring and return to the nest. This is advantageous over continuing with flying as resting in the nest consumes less energy. The second occasion is when an agent is almost dying, and its only chance to survive is to beg for and receive food from a donator. Trust and cheating agents will return to the nest if their energy drops below a certain threshold (4% of maximum energy).

The energy consumption of all bats is based on the decrease in body weight of real vampire bats [5, 9]. The non-linear relation between weight and time after feeding formed the basis for a table with energy consumption values. These values were then tested in the simulation by a group of control agents and scaled to let agents survive for approximately 60 hours without food. This incorporates the difference in energy consumption due to resting in nests. All agents will start within a given nest in the world and are assumed to be females as they have a stable social structure. On average a bat moves to another nest each seven days (as used by [8]). Each nest has the same chance of being chosen and this moving behaviour is simulated by assigning new nest ids to bats after they leave their nest with a probability of  $\frac{1}{7}$ .

### 2.2.2 Behaviour of Trust Agents

Trust agents inherit all the basic behaviours but have the extra ability of trust. The most important aspect of the trust agent is its own limited<sup>2</sup> memory which it can use to store information about other agents in its surroundings. For each agent it can store the *association*, *share rate* and *foraging success*. Only if the thresholds of all these values are met a trust agent will donate food to another agent.

**Association.** For each agent that is in the same nest it will increase the association value for that particular agent in its memory (with 0.3). It will lower the association value by 0.1 of agents that are not in the same nest. This way the relations between agents dynamically degrade or strengthen over time.

**Share Rate.** Trust agents keep track of agents who they donate to and from which they receive food. When they donate food to another agent the share rate will be lowered. The recipient will increase the share rate associated with the donor. A trust agent is optimistic and will donate food to another agent (provided the other constraints are met) if the share rate is greater than zero. When the share rate is precisely zero (no previous sharing interactions) the agent shares with a chance of 0.25.

**Foraging success.** In Wilkinsons simulations a bat didn't share when it's own foraging success was too low. In our simulations finding food is not a binary issue, an agent could find an abundant food source but also one with only a few drops of blood left. So instead of depending on foraging success, the agents shared when they had equal or more than 40% of their total energy level left.

<sup>1</sup>Each agent has a maximum view distance and a two radian ( $\pm 114$  degrees) angle it can detect objects in.

<sup>2</sup>Agents can store up to 20 other agents in their memory. When an unknown agent donates food, the recipient will always remember that agent, regardless the memory limit.

In addition, foraging success is used to assess other agents. Each agent inspects the current food level of the other agents in the nest. This can be seen as a simplified form of grooming which real vampire bats use to assess the amount of blood in the stomach of another bat (as suggested by Wilkinson [7]). Each time an agent inspects the food levels of an agent  $x$  it combines this with information from previous encounters using the following formula:

$$fs(x, t) = 0.8 fs(x, t - 1) + 0.2 fl(x, t) \quad (1)$$

In this formula  $fl(x, t)$  is the current energy of agent  $x$  and  $fs(x, 0) = 0$ . An agent will only share with another agent if this value is equal to or higher than 20% of the maximum energy.

**Begging.** When the energy level of a trust agent drops below 14%<sup>3</sup> of the maximum energy it will beg other agents for food by checking the number of bats in the nest, and approaching them one by one, each only once. An agent only begs agents that are in the same nest and will not approach agents that are flying. If begging is successful the energy levels and memory of both the donor and recipient are updated. If all agents refuse to donate food to an agent it will stop begging for that night.

### 2.2.3 Behaviour of Cheating Agents

Cheating agents include all basic behaviour and are also able to beg for food when running low on energy. The part that makes a cheating agent ‘cheating’ is that it will always refuse to donate food to another agent.

## 2.3 Simulations

The simulations are split into two main parts. The first (‘Life span’) tests the influence of trust on the life span of agents. Each simulation lasted exactly one simulated year (365 simulated days) and always started with 80 agents. Groups of control and trust agents were placed in an environment with different amount of food sources (see Table 1). Then 100 simulations were ran with control agents (resulting in information of 8000 control agents) and 72 with trust agents (5760 agents).

The second block of simulations measures the influence of cheaters. A portion (see Table 2) of the trust agents was replaced with cheating agents (‘Cheaters’) or control agents (‘Cheaters control’). The simulations with trust and control agents are used to determine the influence of the presence of another group. In total 150 simulations were ran.

Table 1: Overview of the simulations in the life span block.

Simulation block Simulation number	Life span						Life span control					
	1	2	3	4	5	6	7	8	9	10	11	12
Number of food sources	5	10	15	20	25	30	5	10	15	20	25	30
Number of trust agents	80	80	80	80	80	80	0	0	0	0	0	0
Number of control agents	0	0	0	0	0	0	80	80	80	80	80	80

Table 2: Overview of the different simulations in the cheaters block.

Simulation block Simulation number	Cheaters					Cheaters control				
	13	14	15	16	17	18	19	20	21	22
Number of food sources	20	20	20	20	20	20	20	20	20	20
Number of trust agents	76	72	60	40	20	76	72	60	40	20
Number of cheating agents	4	8	20	40	60	0	0	0	0	0
Number of control agents	0	0	0	0	0	4	8	20	40	60

## 3 Results

The simulation with only control agents established a base line of performance. For each food source density the average chance of a successful hunt (finding food at night) by a control agent was calculated, these chances were respectively: 26% for 5, 56% for 10, 79% for 15, 88% for 20, 92% for 25 and 95% for 30 food sources.

<sup>3</sup>This is the same as with real vampire bats and translates to about 24 hours of lifetime (including resting).

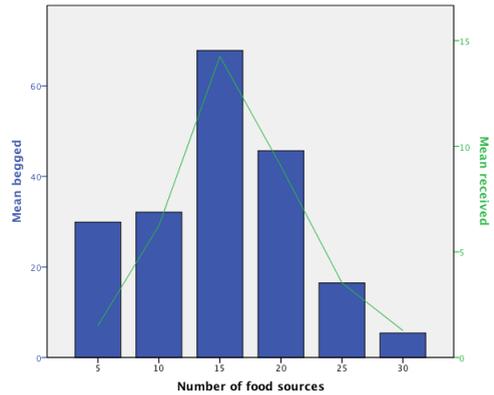
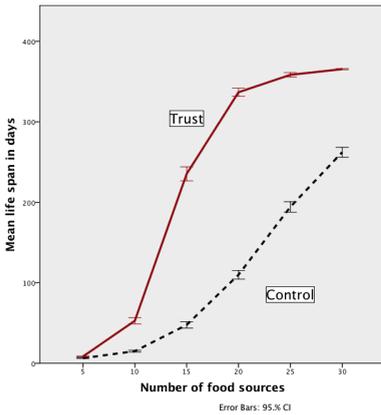


Figure 2: Average life span of control and trust agents Figure 3: Relation between the amount of food in environments versus the amount of food sources. sources and the times trust agents beg (bars) and receive food (line).

### 3.1 Life span

When a trust system is introduced we see the life span increase in comparison to the control agents. Figure 2 and Table 3 show that, except for the low amounts of food, the trust system performs far better than the control system. When the amount of food sources rise we see that the trust system stabilizes a bit due to the time constraint of one year. Also, because agents have more food to their disposal there is less need to beg others for food. This is also confirmed by the frequency of begging and donation, which is lowest at 30 food sources (see Figure 3).

Table 3: Mortality rate of trust and control agents in the life span simulations per food source.

Food sources	5	10	15	20	25	30
Trust agents	100%	99,8%	57,4%	14,8%	4,1%	0,3%
Control agents	100%	100%	99,0 %	85,1%	70,6%	46,3%

As the data is not normally distributed a Mann-Whitney U test has been performed for each combination of food source density between control agents and trust agents. The results of the trust agents differ significantly from the control agents, as seen in Table 4. The effect size is the largest with 15 and 20 food sources, this is supported by Figure 3 which shows that for these food densities agents donate and beg the most.

Table 4: Comparison of the life of control versus trust agents in relation to the amount of food sources.

Food	N		Mean in days		Mean Rank		U	Z	p	Effect size
	Control	Trust	Control	Trust	Control	Trust				
5	880	960	6	8	771,69	1056,91	291445,5	-11,745	0,000	-0,273
10	880	960	15	53	666,89	1152,98	199223,0	-19,624	0,000	-0,457
15	960	960	48	235	597,27	132,73	112098,0	-28,860	0,000	-0,658
20	1760	960	110	337	965,73	2084,24	150008,0	-36,363	0,000	-0,697
25	1760	960	194	358	1036,95	1953,67	275354,0	-31,514	0,000	-0,604
30	1760	960	262	365	1140,37	1764,08	457365,0	-24,430	0,000	-0,468

### 3.2 Cheaters

The influence of control agents on a population of trust agents can be seen in Figure 4. It is clear that trust agents perform better than the control agents but that the control agents, when their number rises, have a negative influence on the population of trust agents. Nonetheless, the difference between trust and control agents is still significant for all the five cases (see Table 5).

When we look at the results of the cheating agents we see that their survival rate is a lot higher than that of the control agents and that their presence influences the trust agents. An overview is shown in Figure 5.

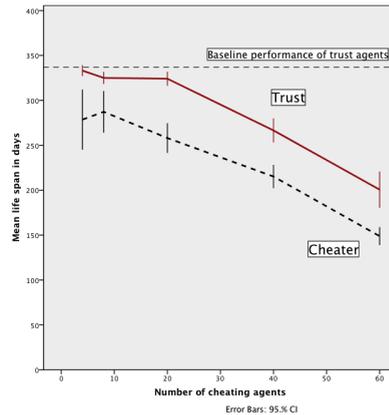
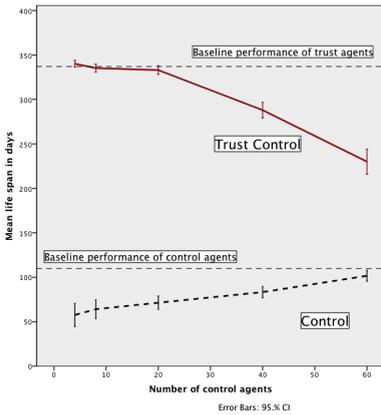


Figure 4: Average life span of control and trust agents Figure 5: Average life span in days of trust and cheating agents placed together in the same simulation.

Table 5: Results of control agents versus trust agents who coexisted together. Although the control agents have a negative influence on the trust agents there is still a large difference between the two groups.

Control agents	N		Mean in days		Mean Rank		U	Z	p	Effect size
	Control	Trust	Control	Trust	Control	Trust				
4	80	1520	58	340	82,92	838,27	3393,5	-20,932	0,000	-0,523
8	160	1440	64	335	127,61	875,27	7537,0	-25,951	0,000	-0,648
20	400	1200	71	333	250,66	938,78	20064,0	-31,597	0,000	-0,789
40	800	800	83	288	501,48	1099,52	80782,5	-26,475	0,000	-0,661
60	1200	400	102	230	702,48	1094,57	122374,0	-14,371	0,000	-0,359

Table 6: Results of cheating and trust agents placed together in the same environment.

Number of cheaters	N		Mean in days		Mean Rank		U	Z	p	Effect size
	Cheat	Trust	Cheat	Trust	Cheat	Trust				
4	40	760	279	333	263,34	407,72	9713,5	-5,612	0,000	-0,183
8	80	720	287	325	322,26	409,19	22541,0	-4,287	0,000	-0,152
20	200	600	258	324	287,39	438,20	37378,0	-9,889	0,000	-0,350
40	400	400	215	267	352,85	448,15	60939,0	-6,096	0,000	-0,216
60	600	200	149	201	380,67	460,00	48100,0	-4,220	0,000	-0,149

Table 7: Life span of trust agents coexisting with cheating agents ('cheat') versus trust agents together with control agents ('control').

Other Agents	N		Mean in days		Mean Rank		U	Z	p	Effect size
	Cheat	Control	Cheat	Control	Cheat	Control				
4	760	1520	333	340	1118,60	1151,45	560958,0	-1,786	0,074	-0,037
8	720	1440	325	335	1039,02	1101,24	488531,5	-3,305	0,001	-0,071
20	600	1200	324	333	881,94	909,78	348861,5	-1,600	0,110	-0,037
40	400	800	267	288	562,22	619,64	144688,0	-3,148	0,002	-0,091
60	200	400	201	230	278,44	311,53	35588,5	-2,277	0,023	-0,092

There seems to be a tipping point around 20 (25%) cheaters or control agents, the average age of trust agents begins to start declining a lot faster after this point. Cheating agents perform, on average, less than trust agents, though better than control agents, and their expected lifespan lowers when their numbers grow. The effect sizes (see Table 6) are not very high (ranging from 0.15 to 0.35), especially in comparison to the effect sizes from the comparison of trust agents with control agents.

When we look at the difference between trust agents living together with few and many cheaters we can compute the effect of a growing cheater population. A Mann-Whitney U test was performed on the means

of the trust agent population in the 4 and 60 cheaters condition. This showed a fairly large effect of  $-0.646$  ( $U = 1547058,0$ ,  $Z = -51,687$  and  $p = 0,000$ ). The effect size between trust agents combined with 5% or 75% control agents is  $-0,428$  ( $U = 166768,0$ ,  $Z = 18,751$  and  $p = 0,000$ ). A comparison of trust agents who live together with cheating agents and trust agents living together with control agents shows differences that are not always significant and have very small effect sizes (see Table 7).

## 4 Discussion

Agents with the ability of trust live significantly longer in comparison to a group without this ability. The size of the effect relates to the amount of food sources, and an optimum lays likely between a 79% and 88% chance of finding food, here agents still have a hard time finding a steady food supply but the trust system can compensate almost completely.

### 4.1 Introducing another group

When we combine trust agents with control agents in the same environment we saw that the life span of trust agents stayed a lot higher than that of the control agents. However, in contrast to a population with only trust agents, there is a decrease. In a second condition control agents were replaced with cheating agents to measure the influence of their cheating behaviour, i.e. beyond their mere presence. The difference between the trust agents and the cheaters remains significant although the effect size is a lot lower than with control agents. Almost all cheaters perform better than control agents. Finally, on average a cheater lives shorter than a trust agent, although, there are cheaters who live equal to or even longer than the trust agents.

For both cases of increasingly added agents, when the numbers of the other agent rise, the average life span of trust agents declines. This can be explained by three (not mutually exclusive) causes:

1. When the number of the other agents increases trust agents have a smaller chance of finding one another and when they find another trust agent it must also be willingly to share some of its food. It is coherent that when the amount of trust agents drops these factors have larger effects.
2. When a trust agent finds food, it will help him, but possibly also another trust agent through a donation. Because a starving bat benefits more from a donation than the donor loses, the food source has the potential to be 'more', in terms of hours of life span, than its initial value. In other words, donating to a fellow trust agent is an investment in the population.
3. From the perspective of the population, it is counterproductive that some agents do not share their food, this is a decrease in the potential availability of the food. But the population suffers even more when some of the collected food is shared to cheaters because there is an additional diminishing of the available food. This donation will not only lower the fitness of the donor, it will also decrease the fitness of the whole population as the energy is 'lost'.

### 4.2 The influence of cheating

The control agents showed us that the mere presence of another group can already influence the life span of trust agents. So what portion of the influence of cheaters is caused by cheating and what by their presence in the world? As cheaters keep all the food they collect for themselves and get some help during rough nights it is pretty straightforward that they have a higher chance of survival than control agents.

It is beneficial for the trust agents when the other agents die as quickly as possible, but due to the longer life span of cheating agents they will use the available food for a longer period than control agents do. Also, because they share more nights with trust agents they have more opportunities to beg for food (and possibly receive some food that would have been better used, from the populations perspective, for starving trust agents).

These two aspects together we can call '*the influence of cheating*' as it is the direct effect on trust agents due to the cheating behaviour. This influence becomes larger when the amount of cheaters rise, the effect size between 5% and 75% cheater is  $-0.646$  where the base level is  $-0.428$  (from the control agents). The effect sizes of the difference between the two groups of trust agents (those living together with cheaters and those that live together with the control agents) are, however, only ranging from  $-0.03$  to  $-0.09$ .

Apparently there is only a small difference between the two groups of trust agents. Cheating agents behave a bit like parasites, they need the trust agents to survive (longer) but their cheating *behaviour* does not have a big influence on the population of trust agents. This is caused by two factors: sharing is not always required and the begging of cheaters is not always successful. Therefore, the absolute quantity of food lost to cheaters is minor.

### 4.3 Conclusion

As the environment wherein the sharing of food was tested is limited there is room for future improvement and research. First of all, there is a structural mismatch between the agents in the simulations and real vampire bats. With a 92% chance of finding food the mortality rate for trust agents is only 4,1% where with real bats this is 24%. So from a biological aspect the simulations can be improved, for example by extending the environment or introducing more vulnerable group members, such as offspring. Such an improvement could also give insight how such a food sharing system emerged and evolved over time.

From a more practical aspect the question arises how we could use our gained insights of the food sharing system for practical applications. As the system is decentralised and able to create associations without prior knowledge, a possible application is that of (wireless) mesh networks. As these networks require users to share resources (traffic through their node) a trust system like that of the vampire bat can be used to decide whom to share with. Such an implementation will, however, need more research in the effect of cheaters, something which lays outside the scope of this paper.

To conclude, the trust system of the vampire bat is a simple but effective decentralised sharing system that is able to elevate the fitness of a group of agents that use it. Agents get to know each other and make decisions who to share with. The system works best when there is some deficiency in food supply. While cheaters do have some influence, if their numbers does not get out of hand, trusters should be fine.

## References

- [1] A. H. Harcourt. Help, cooperation and trust in animals. In *Cooperation and prosocial behaviour*, pages 15–26. Cambridge University Press, 1991.
- [2] J. Klein. Breve: a 3D environment for the simulation of decentralized systems and artificial life. In *Proceedings of the eighth international conference on Artificial life*, pages 329–334, 2003.
- [3] L. Spector, J. Klein, C. Perry, and M. Feinstein. Emergence of collective behavior in evolving populations of flying agents. In *Genetic and Evolutionary Computation—GECCO 2003*, pages 200–200. Springer, 2003.
- [4] R. L. Trivers. The evolution of reciprocal altruism. *The Quarterly Review of Biolog*, 46(1):35–57, March 1971.
- [5] G. S. Wilkinson. Reciprocal food sharing in the vampire bat. *Nature*, 308:181–184, March 1984.
- [6] G. S. Wilkinson. The social organization of the common vampire bat. *Behavioral Ecology and Sociobiology*, 17(2):123–134, 1985.
- [7] G. S. Wilkinson. Social grooming in the common vampire bat, *desmodus rotundus*. *Animal Behaviour*, 34(6):1880–1889, 1986.
- [8] G. S. Wilkinson. Reciprocal altruism in bats and other mammals. *Ethology and Sociobiology*, 9:85–100, 1988.
- [9] G. S. Wilkinson. Food sharing in vampire bats. *Scientific American*, 262(2):64–70, 1990.
- [10] W. A. Wimsatt. Transient behavior, nocturnal activity patterns, and feeding efficiency of vampire bats (*desmodus rotundus*) under natural conditions. *Journal of Mammalogy*, 50(2):pp. 233–244, 1969.

# On extended conflict-freeness in argumentation

Martin Caminada<sup>1</sup>

Srdjan Vesic<sup>2</sup>

*University of Luxembourg  
6 rue Richard Coudenhove-Kalergi, L-1359 Luxembourg  
{martin.caminada,srdjan.vesic}@uni.lu*

## Abstract

This paper studies a possibility to represent  $n$ -ary conflicts within an argumentation framework having only binary attacks. We show that different instantiations of the abstract argumentation framework defined by Dung use very similar constructs for dealing with  $n$ -ary conflicts. We start by studying this procedure on two fully-instantiated systems from the argumentation literature and then show that it can also be formalised on the abstract level. We argue that this way of handling  $n$ -ary conflicts has two benefits. First, it allows to represent all the information within a standard argumentation framework, only by using arguments and attacks (e.g. without adding a new component to store the sets of conflicts). Second, all the added arguments have an intuitive interpretation, i.e. their meaning on the instantiated and on the abstract level is conceptually clear.

## 1 Introduction

The field of formal argumentation [6, 14] is based on the idea that reasoning can be performed by constructing and evaluating arguments, which are composed of a number of reasons that together support a claim. Arguments distinguish themselves from proofs by the fact that they are defeasible, that is, the validity of their conclusions can be disputed by other arguments. Whether a claim can be accepted therefore depends not only on the existence of an argument that supports this claim, but also on the existence of possible counter arguments, that can then themselves be attacked by counter arguments, etc.

This approach to reasoning has drawn a significant amount of attention since the conceptualisations of Pollock [12, 13], Vreeswijk [18], and Simari and Loui [16]. One of the common features of some of the formalisations in the 1990s (e.g. the work of Baroni et al. [4] or by Vreeswijk [18]) is the possibility to explicitly represent collective attacks between arguments. For example, in those frameworks, one is able to specify that there exist a set of three arguments  $\{A, B, C\}$  such that neither  $A$  nor  $B$  attack  $C$ , but  $A$  and  $B$  together attack  $C$ .

Nowadays, much research on the topic of argumentation is based on the theory proposed by Dung [10]. It allows one to abstract from the origin and the structure of arguments, by representing an argumentation framework as a directed graph, whose vertices correspond to arguments and arcs to attacks between them. However, this attack relation is binary, and it is not possible to explicitly specify  $n$ -ary attacks for  $n \geq 3$ . For instance, it is not possible to specify the existence of three arguments  $\{A, B, C\}$  such that neither  $A$  nor  $B$  attack  $C$ , but  $A$  and  $B$  together attack  $C$ .

Does this mean that it is impossible to represent ternary conflicts in such a setting? Are there instantiations of Dung's abstract theory that make it possible to specify that a set of arguments should not be accepted even if it is conflict-free with respect to a binary attack relation, i.e. even if there exist no arguments  $A, B$  in that set such that  $A$  attacks  $B$ ?

Take for example three different formulae  $\varphi, \psi, \omega$  such that the union of any two of those formulae is consistent and the union of all three formulae is inconsistent. Furthermore, let argument  $A$  be built by

<sup>1</sup>Research of MC was supported by the National Research Fund, Luxembourg (LAAMI project).

<sup>2</sup>SV was funded by the National Research Fund, Luxembourg. His work on this paper was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. This Programme is supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

using only  $\varphi$ ,  $B$  by using only  $\psi$  and  $C$  by using only  $\omega$ . In virtually all instantiations of Dung's abstract argumentation theory, set  $\{A, B, C\}$  is conflict-free. However, there are ways to make sure that this set never appears as a part of an extension. The goal of this paper is to show that different instantiations of Dung's theory use very similar techniques to deal with this problem, which we refer to as "extended conflict-freeness". We will also argue that the technique used has two positive features. First, it allows to represent all the information about conflicts within an argumentation framework, without adding new components (such as a Boolean formula to represent a constraint [9] or a formula representing an acceptance condition for every argument [7]). Second, the added arguments have an intuitive interpretation, i.e. their meaning on the instantiated and on the abstract level is conceptually clear.

This paper is organised as follows: Section 2 defines the notions from argumentation theory needed for the present paper, Section 3 shows that different instantiations of Dung's theory from the argumentation literature use the same way to deal with extended conflict-freeness, Section 4 formalises this mechanism on the abstract level and Section 5 concludes.

## 2 Preliminaries

An argumentation framework is defined as a binary oriented graph, whose nodes represent arguments and whose arcs represent the attacks between them [10].

**Definition 1** An argumentation framework (AF) is a pair  $\mathcal{F} = (\mathcal{A}, \mathcal{R})$ , where  $\mathcal{A}$  is a set of arguments and  $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$  is a binary relation representing the attacks between the arguments. For two arguments  $a, b \in \mathcal{A}$ , the notation  $a\mathcal{R}b$  or  $(a, b) \in \mathcal{R}$  means that  $a$  attacks  $b$ .

The central notions in this theory are: conflict-freeness, defence and admissibility.

**Definition 2** Let  $\mathcal{F} = (\mathcal{A}, \mathcal{R})$  be an argumentation framework and  $S \subseteq \mathcal{A}$  and  $a \in \mathcal{A}$ .

- $S$  is conflict-free if and only if there is no  $a, b \in S$  such that  $a\mathcal{R}b$ .
- $S$  defends argument  $a$  if and only if for every  $b \in \mathcal{A}$  if  $b\mathcal{R}a$  then there exists  $c \in S$  such that  $c\mathcal{R}b$ .
- $S$  is an admissible set in  $\mathcal{F}$  if and only if  $S$  is conflict-free and defends all its elements.

A semantics is a function which, given an argumentation framework, calculates the sets of arguments which can be accepted together, called *extensions*. Let us now define some of the most commonly used semantics.

**Definition 3** Let  $\mathcal{F} = (\mathcal{A}, \mathcal{R})$  be an AF and  $S \subseteq \mathcal{A}$ . We say that a set  $S$  is admissible if and only if it is conflict-free and defends all its elements.

- $S$  is a complete extension if and only if  $S$  defends all its arguments and contains all the arguments it defends.
- $S$  is a preferred extension if and only if it is a maximal (with respect to set inclusion) admissible set.
- $S$  is a stable extension if and only if  $S$  is conflict-free and for all  $a \in \mathcal{A} \setminus S$ , there exists  $b \in S$  such that  $b\mathcal{R}a$ .
- $S$  is a semi-stable extension if and only if  $S$  is a complete extension and the union of the set  $S$  and the set of all arguments attacked by  $S$  is maximal (for set inclusion).
- $S$  is a grounded extension if and only if  $S$  is a minimal (for set inclusion) complete extension.
- $S$  is an ideal extension if and only if  $S$  is a maximal (for set inclusion) admissible set contained in every preferred extension.

**Definition 4** A semantics  $\sigma$  is admissibility-based if and only if for every argumentation framework  $\mathcal{F} = (\mathcal{A}, \mathcal{R})$  it holds that every extension of  $\mathcal{F}$  under semantics  $\sigma$  is an admissible set.

**Example 1** Complete, preferred, stable, semi-stable, grounded and ideal semantics are all admissibility-based.

### 3 Extended conflict-freeness in instantiated argumentation frameworks

In this section, we study the central question of the paper: how are  $n$ -ary conflicts handled in argumentation for  $n \geq 3$ ? Let us start with two examples. We start by examining two well-known approaches to instantiated argumentation: so-called “logic-based” approach, based on classical propositional logic [5], and so-called “rule-based” approach, which does not make use of propositional logic, but instead constructs arguments from rules of a given defeasible theory [8].

**Example 2** Suppose an instantiation of Dung’s theory with propositional logic where arguments are pairs (support, conclusion), support being a minimal consistent set of propositional formulae and conclusion being a formula such that support  $\vdash$  conclusion, where  $\vdash$  is the consequence operator from classical propositional logic [5]. Suppose a knowledge base  $\Sigma = \{x, y, \neg x \vee \neg y\}$ . Let  $A_1 = (\{x\}, x)$ ,  $A_2 = (\{y\}, y)$  and  $A_3 = (\{\neg x \vee \neg y\}, \neg x \vee \neg y)$  be three arguments and let  $S = \{A_1, A_2, A_3\}$ . Virtually all attack relations used in this setting satisfy conflict-dependence [1], that is, if an argument attacks another one, then the union of their supports is inconsistent. Furthermore, for any conflict-dependent relation, neither  $A_1$  attacks  $A_2$  nor  $A_2$  attacks  $A_1$ . The same holds for other pairs:  $A_1, A_3$  and  $A_2, A_3$ . Thus,  $S$  is conflict-free. However, under most of the existing semantics, one would like to ensure that no extension contains set  $S$ .

**Example 3** Suppose the ASPIC instantiation of Dung’s theory [8], where arguments are built from strict and defeasible rules. Suppose a defeasible theory  $(S, \mathcal{D})$ , with  $S = \{x, y \rightarrow \neg z; z, y \rightarrow \neg x; x, z \rightarrow \neg y\}$  and  $\mathcal{D} = \{\Rightarrow x; \Rightarrow y; \Rightarrow z\}$ . Let  $A_1 = (\Rightarrow x)$ ,  $A_2 = (\Rightarrow y)$  and  $A_3 = (\Rightarrow z)$  be three arguments and let  $S = \{A_1, A_2, A_3\}$ . Set  $S$  is conflict-free in this framework. However, one would like to avoid having an extension  $\mathcal{E}$  such that  $S \subseteq \mathcal{E}$ .

Both instantiations [5, 8] of Dung’s abstract theory mentioned in the previous examples ensure consistency by *constructing additional arguments*. How is this achieved? The answer is that one has to make sure that all the relevant arguments are constructed. The frameworks from Example 2 and 3 are not complete [17, Section 3] in the sense that not all arguments that can be constructed from the available knowledge are in the framework. Let us show how adding arguments solves the problem of modelling extended conflict-freeness with a binary attack relation.

**Example 4 (Example 2 Cont.)** The intuition in this example is that  $A_1$ ,  $A_2$  and  $A_3$  are not acceptable together: To be able to express this, one needs to create more arguments. We will construct an argument  $B_3$ , telling “ $A_1$  and  $A_2$  are in the extension, so  $A_3$  cannot be in the extension”. In other words, let  $B_3 = (\{x, y\}, \neg(\neg x \vee \neg y))$ . We can construct two more arguments telling that  $A_1$  and (respectively  $A_2$ ) cannot be in the extension since  $A_2$  and  $A_3$  (respectively  $A_1$  and  $A_3$ ) are in the extension:  $B_1 = (\{y, \neg x \vee \neg y\}, \neg x)$ ,  $B_2 = (\{x, \neg x \vee \neg y\}, \neg y)$ . Let us suppose that an argument  $X$  attacks argument  $Y$  if and only if there exists a formula  $\varphi$  in the support of  $Y$  such that the conclusion of  $X$  is logically equivalent to  $\neg\varphi$ . The corresponding argumentation graph is shown in Figure 1. This argumentation framework has four complete extensions  $\mathcal{E}_1 = \emptyset$ ,  $\mathcal{E}_2 = \{B_3, A_1, A_2\}$ ,  $\mathcal{E}_3 = \{B_1, A_2, A_3\}$  and  $\mathcal{E}_4 = \{B_2, A_1, A_3\}$ . There are three preferred extensions (that are also stable and semi-stable):  $\mathcal{E}_2$ ,  $\mathcal{E}_3$  and  $\mathcal{E}_4$ . The grounded extension coincides with the ideal extension and is equal to  $\mathcal{E}_1 = \emptyset$ . The main point is that “additional” arguments  $B_1, B_2, B_3$  allow the ternary conflict to be encoded in the argumentation graph.

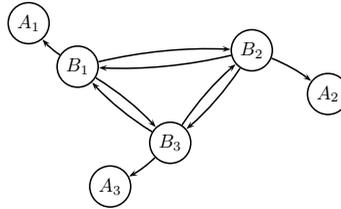


Figure 1: Arguments  $B_1$ ,  $B_2$  and  $B_3$  ensuring extended conflict-freeness

Note that other arguments can be constructed in Example 4, but they are not essential for the present discussion. Indeed, an infinite number of arguments can be constructed in this formalism, but it is known [2] that in this setting, for every infinite argumentation framework built from a finite number of propositional formulae, there exists an equivalent finite framework.

Let us see what happens in ASPIC [8] instantiation of Dung's theory.

**Example 5 (Example 3 Cont.)** Let  $B_3 = (A_1, A_2 \rightarrow \neg z)$ ,  $B_1 = (A_2, A_3 \rightarrow \neg x)$ , and  $B_2 = (A_1, A_3 \rightarrow \neg y)$ . The argument graph corresponding to this formalisation is the same as the graph from Example 4 (depicted in Figure 1).

Examples 4 and 5 show that the notion of extended conflict-freeness is already present in the literature, though not explicitly. It shows that existing instantiations construct argumentation framework in which the information about  $n$ -ary conflicts is encoded using a particular pattern in the graph.

## 4 Extended conflict-freeness on the abstract level

In this section, we abstract from the structure and contents of arguments and generalise the ideas presented in the previous section. Our goal is to show that every instantiation of Dung's abstract theory [10] can benefit from the same pattern which allows to deal with  $n$ -ary conflicts. We suppose that one is given an argumentation framework  $\mathcal{F} = (\mathcal{A}, \mathcal{R})$  and a collection  $S_1, \dots, S_n \subseteq \mathcal{A}$  of sets such that each set  $S_i$  represents a minimal conflict, in the sense that one does not want any extension to contain any of  $S_i$ -s. Note that this paper does not study the question *how to identify sets*  $S_1, \dots, S_n$ . That question cannot be solved on the abstract level. We only identify and study a mechanism which ensures that extensions do not contain any of those sets, once the collection  $S_1, \dots, S_n$  is known. This is reflected in the next definition, which just accepts *any* collection of sets  $S_1, \dots, S_n \subseteq \mathcal{A}$ . Of course, the question how to identify  $S_1, \dots, S_n$  is a very important one, but is not a topic of this paper.

**Definition 5** Let  $\mathcal{F} = (\mathcal{A}, \mathcal{R})$  be an argumentation framework. A collection of minimal argumentation conflicts is a finite collection of sets  $\mathcal{C} = \{S_1, \dots, S_n\}$  such that for all  $i \in \{1, \dots, n\}$ , we have  $S_i \subseteq \mathcal{A}$ .

We now present on the abstract level a way used to ensure extended conflict freeness in many extended instantiations of Dung's theory. The idea is that for every minimal argumentation conflict  $S_i = \{A_1^i, \dots, A_{k_i}^i\}$ , one generates the additional arguments  $B_1^i, \dots, B_{k_i}^i$  such that

- each  $B_k^i$  attacks  $A_k^i$ ,
- each  $B_j^i$  also attacks every  $B_k^i$ , for  $j \neq k$ ,
- each attacker of  $A_k^i$  also attacks every  $B_j^i$ , for  $j \neq k$ .

**Example 6** To illustrate this idea, consider an argumentation framework  $\mathcal{F} = (\mathcal{A}, \mathcal{R})$  with  $\mathcal{A} = \{A_1, A_2, A_3, A_4, A_5, C\}$ ,  $\mathcal{R} = \{(C, A_1)\}$  and with minimal conflicts  $S_1 = \{A_1, A_2, A_3\}$  and  $S_2 = \{A_2, A_4, A_5\}$ . The extended framework  $\mathcal{F}' = (\mathcal{A}', \mathcal{R}')$ , with added arguments to model those conflicts, is depicted in Figure 2. Arguments  $B_1, B_2$  and  $B_3$  correspond to conflict  $S_1$ , whereas arguments  $B_2, B_4$  and  $B_5$  refer to  $S_2$ . Since  $C$  attacks  $A_1$  in the original framework, then it also attacks  $B_2, B_3, B_4$  and  $B_5$ .

The next definition formalises this procedure.

**Definition 6** Let  $\mathcal{F} = (\mathcal{A}, \mathcal{R})$  be an argumentation framework, let  $\sigma$  be an admissibility-based semantics and let  $\mathcal{C} = \{S_1, \dots, S_n\}$  be a collection of minimal argumentation conflicts. Let  $S_1 = \{A_1^1, \dots, A_{k_1}^1\}, \dots, S_i = \{A_1^i, \dots, A_{k_i}^i\} \dots S_n = \{A_1^n, \dots, A_{k_n}^n\}$ .

The extended conflict-free version of  $\mathcal{F}$  with respect to  $\mathcal{C}$  is defined as  $\mathcal{F}' = (\mathcal{A}', \mathcal{R}')$ , where:

- $\mathcal{A}' = \mathcal{A} \cup \mathcal{B}_1 \cup \dots \cup \mathcal{B}_n$ , where  $\mathcal{B}_1 = \{B_1^1, \dots, B_{k_1}^1\}, \dots, \mathcal{B}_i = \{B_1^i, \dots, B_{k_i}^i\} \dots \mathcal{B}_n = \{B_1^n, \dots, B_{k_n}^n\}$ .
- $\mathcal{R}' = \mathcal{R} \cup \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$ , where<sup>1</sup>
  - $\mathcal{R}_1 = \{(B_j^i, A_j^i) \mid i \in \{1, \dots, n\}, j \in \{1, \dots, k_i\}\}$ ,
  - $\mathcal{R}_2 = \{(B_j^i, B_l^i) \mid i \in \{1, \dots, n\}, j, l \in \{1, \dots, k_i\}, j \neq l\}$
  - $\mathcal{R}_3 = \{(C, B_j^i) \mid i \in \{1, \dots, n\}, j, l \in \{1, \dots, k_i\}, C \in \mathcal{A}, (C, A_j^i) \in \mathcal{R} \text{ and } j \neq l\}$ .

<sup>1</sup>Note that the values of  $j$  and  $l$  depend on  $i$ , i.e.  $j = j(i)$  and  $l = l(i)$ .

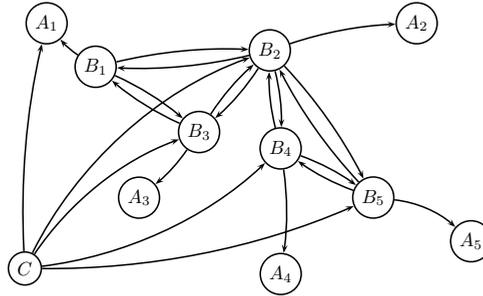


Figure 2: Extension of argumentation framework  $\mathcal{F}$  from Example 6 with auxiliary arguments and attacks

The previous definition shows how  $n$ -ary conflicts are translated into binary conflicts in various instantiations of Dung’s theory.

Let us analyse this in detail. Arguments  $B_j^i$  are added to ensure consistency at the instantiated level. An argument  $B_j^i$  stands for: “there is a conflict  $S_i$ , and if you want to accept  $A_1^i, \dots, A_{j-1}^i, A_{j+1}^i, \dots, A_{k_i}^i$ , then you cannot accept  $A_j^i$ ”. To achieve this,  $B_j^i$  attacks  $A_j^i$ . Also, for every  $i$ , arguments  $B_1^i, \dots, B_{k_i}^i$  are mutually incompatible, since each of them relies on all but one arguments from a minimal argumentation conflict. Finally, what is the use of adding attacks from “ $C$  arguments”, i.e. from existing arguments to “ $B$  arguments”? The point is that we do not want to destroy existing incompatibilities between arguments. As an illustration, there may be an accepted attacker  $C$  of  $A_1$  in Example 3. In such a situation, one should just reject  $A_1$  and accept  $A_2$  and  $A_3$ . Let us construct such an example.

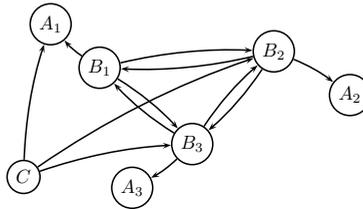


Figure 3: Argument  $C$  solving a ternary conflict: a correct solution

**Example 7 (Example 3 Contd.)** Let us suppose that the defeasible theory  $\langle S, \mathcal{D} \rangle$ , is updated. Thus, we obtain a new theory  $\langle S', \mathcal{D}' \rangle$ , with  $S' = S \cup \{\rightarrow t; t \rightarrow \neg[\Rightarrow x]\}$  and  $\mathcal{D}' = \mathcal{D}$ . Here,  $t$  is a fact such that  $x$  can be defeasibly concluded only in absence of  $t$ , i.e.  $t$  is an undercut of the defeasible rule allowing to conclude  $x$ . In addition to  $A_1, A_2, A_3, B_1, B_2, B_3$ , another significant argument can be constructed, namely  $C = ((\rightarrow t) \rightarrow \neg[\Rightarrow x])$ . The attack graph is depicted in Figure 3. There is a unique complete / stable / semi-stable / preferred / grounded / ideal extension:  $\{C, B_1, A_2, A_3\}$ . Intuitively, since  $A_1$  is undercut by  $C$  (which is undefeated) then  $A_1$  should not be accepted. So, there is no reason not to accept both  $A_2$  and  $A_3$ . In such a situation, forgetting to add attacks  $(C, B_2) \in \mathcal{R}'$  and  $(C, B_3) \in \mathcal{R}'$ , would result in a non-intuitive solution, as shown in Figure 4. The argumentation framework from Figure 4 has four complete extensions  $\mathcal{E}_1 = \{C\}$ ,  $\mathcal{E}_2 = \{C, B_1, A_2, A_3\}$ ,  $\mathcal{E}_3 = \{C, B_2, A_3\}$  and  $\mathcal{E}_4 = \{C, B_3, A_2\}$ . There are three stable / semi-stable / preferred extensions, namely  $\mathcal{E}_2, \mathcal{E}_3$  and  $\mathcal{E}_4$ . The grounded extension coincides with the ideal extension and is equal to  $\mathcal{E}_1$ . Note that  $B_2$  stands for: “if you accept  $A_1$  and  $A_3$ , then you cannot

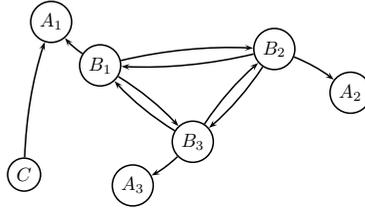


Figure 4: Argument  $C$  solving a ternary conflict: an incorrect solution

accept  $A_2$ ". Informally, the conclusion of  $B_2$  is only valid in a situation when one does accept both  $A_1$  and  $A_3$ . But, since  $A_1$  is not accepted (because of  $C$ ), then arguments  $B_2$  and  $B_3$  "do not make sense": as soon as  $C$  is accepted, they should be rejected. Also, this produces two undesirable extensions:  $\mathcal{E}_3$  and  $\mathcal{E}_4$ . Namely, there is no reason to accept  $A_2$  and not  $A_3$  or vice versa, since there is no incompatibility whatsoever between  $A_2$  and  $A_3$ .

The next theorem proves that the extensions of the extended conflict-free version of every argumentation framework satisfy extended conflict-freeness. Let us first formally define this notion.

**Definition 7** Let  $\mathcal{F} = (\mathcal{A}, \mathcal{R})$  be an argumentation framework, let  $\sigma$  be an admissibility-based semantics and let  $\mathcal{C} = \{S_1, \dots, S_n\}$  be a collection of minimal argumentation conflicts. A set  $S \subseteq \mathcal{A}$  satisfies extended-conflict-freeness if and only if there exists no  $S_i \in \mathcal{C}$  such that  $S_i \subseteq S$ .

**Theorem 1** Let  $\mathcal{F} = (\mathcal{A}, \mathcal{R})$  be an argumentation framework, let  $\sigma$  be an admissibility-based semantics and let  $\mathcal{C} = \{S_1, \dots, S_n\}$  be a collection of minimal argumentation conflicts. Let  $\mathcal{F}' = (\mathcal{A}', \mathcal{R}')$  be the extended conflict-free version of  $\mathcal{F}$ . Then: every extension of  $\mathcal{F}'$  under  $\sigma$  satisfies extended-conflict-freeness.

**Proof** Let  $S_1 = \{A_1^1, \dots, A_{k_1}^1\}, \dots, S_i = \{A_1^i, \dots, A_{k_i}^i\} \dots, S_n = \{A_1^n, \dots, A_{k_n}^n\}$ . We will show that there is no extension containing a set  $S_i$ . To prove the theorem by reductio ad absurdum, suppose the contrary, i.e. let  $\mathcal{E}$  be an extension of  $\mathcal{F}'$  under semantics  $\sigma$ , let  $i \in \{1, \dots, n\}$  and let  $S_i \subseteq \mathcal{E}$ . Since  $\sigma$  is an admissibility-based semantics, then  $\mathcal{E}$  is an admissible set. This means that for every  $j \in \{1, \dots, k_i\}$ ,  $B_j^i \notin \mathcal{E}$ . Again from admissibility of  $\mathcal{E}$ , since for every  $j \in \{1, \dots, k_i\}$ , we have  $(B_j^i, A_j^i) \in \mathcal{R}'$ , then for every  $j \in \{1, \dots, k_i\}$  there exists  $C_j \in \mathcal{A}$  such that  $(C_j, B_j^i) \in \mathcal{R}'$  and  $C_j \in \mathcal{E}$  (informally speaking, arguments of "type  $B$ " can only be attacked by arguments of "type  $C$ "). Note that the case  $|S_i| = 1$  is not possible, since that would mean that  $B_1^i$  is not attacked in  $\mathcal{F}'$ . Thus,  $|S_i| \geq 2$ . This means that there exists  $A_2^i \in \mathcal{E}$ . From Definition 6, we have that  $(C_1, A_2^i) \in \mathcal{R}'$ . This would mean that set  $\mathcal{E}$  is not conflict-free. Contradiction with the facts  $A_2^i \in \mathcal{E}$ ,  $C_1 \in \mathcal{E}$  and that  $\mathcal{E}$  is an admissible set. This means that no extension contains a set  $S_i \in \mathcal{C}$ . ■

## 5 Discussion, related literature and future work

In this paper, we showed how  $n$ -ary conflicts with  $n \geq 3$  are dealt with in different instantiations of Dung's abstract argumentation theory. To the best of our knowledge, this is the first paper pointing out that different instantiations of Dung's theory use very similar techniques to deal with this issue. We believe that this is one of the rare modelisations which at the same time allows to represent all the information about conflicts within an argumentation framework, without adding new components (e.g. a Boolean formula to represent a constraint or a formula representing an acceptance condition for every argument) and where the added arguments have an intuitive interpretation, i.e. their meaning on the instantiated and on abstract level is conceptually clear. In this section, we review the related work and show why we believe that some existing formalisations violate at least one of those two positive properties.

Brewka and Woltran [7] introduced *abstract dialectical frameworks* (ADFs), a generalisation of Dung's theory. They argue that the only interaction between arguments in Dung's framework is attack, and propose to attach to every argument an acceptance condition in form of a classical propositional logic formula,

using other arguments as atoms. The fact that  $A_1$  and  $A_2$  together attack  $A_3$  can be modelled by attaching the formula  $\neg(A_1 \wedge A_2)$  as the acceptance condition of argument  $A_3$ . They prove that for every ADF, there exists a Dung-style argumentation framework such that any *model* of the original ADF corresponds to a *stable extension* of the corresponding AF and vice versa. They also show that similar translations are possible between the *well-founded model* of an ADF and the *grounded extension* of an AF and between *stable models* of an ADF *stable extensions* of an AF. It is clear that ADFs are a more general tool (e.g. they also allow to represent supports) than the approach described in this paper. However, the approach we present can be used for every admissibility-based semantics without change, whereas the translation from an ADF to an AF was proved only for three semantics. Also, it is intuitively clear what is the meaning of every argument  $B_j^i$  that is added to the framework and which arguments should attack it, whereas when an ADF is translated to an AF, some arguments have purely technical meaning. A part of future work will also be to compare the robustness of the approach where an ADF is translated to an AF to represent  $n$ -ary attacks and the approach reported in the present paper. Namely, we want to explore how well the *dynamics of argumentation* is handled by the two approaches. This includes the capacity of an approach to be updated when a new  $n$ -ary attack is added, without having to recompute everything. We will also formally study the possibility of using constrained argumentation frameworks [9] for modelling  $n$ -ary attacks and compare those results with the results obtained by the approach described in the present paper. Once again, a benefit of using the present approach is that it is able to express all the conflicts on the basic AF level, whereas it generates only arguments with a clear conceptual meaning, i.e.  $A_i^j$  stands for “argument  $A_i$  cannot be accepted because all arguments  $A_1, \dots, A_{i-1}, A_{i+1}, \dots, A_n$  are already accepted”.

Another related paper is the work [15] aimed at constructing *argumentation patterns*, such as conjunction, disjunction, or more complex constructs (e.g. Toulmin scheme). While the present paper’s goal is to understand how existing instantiated systems deal with  $n$ -ary attacks, the work of Villata et al. deals with situations where argumentation frameworks are not generated from a knowledge base, but where the knowledge engineer has to directly design arguments and attacks.

An important remark is that this paper shows the similarity between so called *logic-based instantiations* [3, 11] and *rule-based instantiations* [8] when it comes to generating particular patterns in the argumentation framework (as illustrated by Examples 4 and 5). Our future work will be to study *why* those patterns occur, are there other possibilities to model  $n$ -ary conflicts, and if yes, what is the “best” way to do it.

This paper presents the first step towards understanding how extended conflict-freeness is handled in instantiations of Dung’s framework. A part of our future work is to continue the formalisation and to prove its properties. For instance, we argue in Example 7 that  $C$  should attack  $B_2$  and  $B_3$  and explain the intuition behind such a definition. We plan to formalise these explanations and prove what kind of properties the presented formalisation satisfies in general.

## References

- [1] L. Amgoud and Ph. Besnard. Bridging the gap between abstract argumentation systems and logic. In *International Conference on Scalable Uncertainty Management (SUM’09)*, pages 12–27, 2009.
- [2] L. Amgoud, Ph. Besnard, and S. Vesic. Identifying the core of logic-based argumentation systems. In *23rd International Conference on Tools with Artificial Intelligence (ICTAI’11)*, pages 633–636, 2011.
- [3] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34:197–216, 2002.
- [4] P. Baroni, M. Giacomin, and G. Guida. Extending abstract argumentation systems theory. *Artificial Intelligence Journal*, 120(2):251–270, 2000.
- [5] Ph. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence Journal*, 128 (1-2):203–235, 2001.
- [6] Ph. Besnard and A. Hunter. *Elements of Argumentation*. MIT Press, 2008.
- [7] G. Brewka and S. Woltran. Abstract dialectical frameworks. In *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR’10)*. AAAI Press, 2010.

- [8] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence Journal*, 171 (5-6):286–310, 2007.
- [9] S. Coste-Marquis, C. Devred, and P. Marquis. Constrained argumentation frameworks. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR '06)*, pages 112–122, 2006.
- [10] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and  $n$ -person games. *Artificial Intelligence Journal*, 77:321–357, 1995.
- [11] N. Gorogiannis and A. Hunter. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artificial Intelligence Journal*, 175:1479–1497, 2011.
- [12] J. L. Pollock. How to reason defeasibly. *Artificial Intelligence Journal*, 57:1–42, 1992.
- [13] J. L. Pollock. *Cognitive Carpentry. A Blueprint for How to Build a Person*. MIT Press, Cambridge, MA, 1995.
- [14] I. Rahwan and G. Simari, editors. *Argumentation in Artificial Intelligence*. Springer, 2009.
- [15] S. Villata S, G. Boella, and L. van der Torre. Argumentation patterns. In *8th International Workshop on Argumentation in Multi-Agent Systems (ArgMAS'11)*, pages 133–150, 2011.
- [16] G.R. Simari and R.P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence Journal*, 53:125–157, 1992.
- [17] S. Vesic. Maxi-consistent operators in argumentation. In *20th European Conference on Artificial Intelligence (ECAI'12)*, 2012.
- [18] G. A. W. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence Journal*, 90:225–279, 1997.

# A Nonparametric Evaluation of SysML-based Mechatronic Conceptual Design

Mohammad Chami <sup>a</sup>      Haitham Bou Ammar <sup>b</sup>      Holger Voos <sup>c</sup>      Karl Tuyls <sup>b</sup>

Gerhard Weiss <sup>b</sup>

<sup>a</sup> *Institute of Applied Research, Uni. of Appl. Sciences Ravensburg-Weingarten, Germany*

<sup>b</sup> *Department of Knowledge Engineering, Maastricht University, The Netherlands*

<sup>c</sup> *Research Unit in Engineering Science, University of Luxembourg, Luxembourg*

## Abstract

Mechatronic technologies are used in a wide range of industries, from aerospace to automotive, manufacturing and even to personal devices, such as cd/dvd players. Although their multidisciplinary nature provides great functionalities, it is still one of the substantial challenges which frequently impede their design process. Apart from this problem, an early system design evaluation while adhering to adaptable design requirements is still missing. In this paper we propose a SysML-based method for an Intelligent Conceptual Design Evaluation of mechatronic systems, abbreviated as SysDICE. Particularly, we contribute by, firstly, making use of SysML as a common modeling language for the engineering team involved in the design process and secondly, by adopting a widely used, in artificial intelligence, pattern recognition tool, namely non-parametric regression, to support a multi-alternative design mechanism, with the aim of attaining the best combination of components' alternatives that suits a set of prioritized numerical requirements. To evaluate our framework, we have conducted two design experiments: (1) a two-wheel differential drive robot, and (2) a quad-rotor unmanned aerial vehicle. Results prove how our framework can assist system engineers and support the design process.

## 1 Introduction

The system design phase of mechatronics typically exhibits a multidisciplinary nature by aggregating various engineering disciplines (i.e., mechanical, electrical, software and control), project and business management fields. From a system engineering perspective [1], system engineers must have a broad knowledge on the system from the end-user domain till the system's technical engineering domains. This nature imposes a substantial challenge that deals with integrating the involved human factors with their methodologies, modeling languages and software tools for the aim of attaining an efficient system design.

In theory the course of system design from idea creation to product disposal has been successfully proposed (e.g. [2, 3]). However, the industrial development techniques are still mono-disciplinary [4]. Particularly, the integration phase of the different disciplines' outcomes yet arrive at later stages, which makes the procedure expensive, cost and time in-efficient. Moreover, in reality, a document-based manner has been followed to hold the disciplines' interdependencies (i.e., how, when and in what way any discipline influences another). This frequently leads to weak synchronization between the interdependencies of entities and can result in inefficiencies that often appear during the integration and testing [1]. Therefore, an early integrated evaluation of the system, as a whole, is strongly demanded.

So far, little attention has been given to the collaborative work for evaluating designs in a sequel of making the procedure adaptable, efficient, and intelligent. In this paper we target the previously mentioned problems and contribute by: (1) capturing the interdisciplinary information across system engineers and designers using SysML to generate a system design model, which is (2) mathematically formulated for the aim of the satisfaction of a set of prioritized numerical requirements by (3) adopting non-parametric regression. In this way the system design procedure is more adaptable and coherent.

## 2 Related Work

Multidisciplinary approaches in mechatronic design have been frequently discussed in research. For instance, in [5] a high-level system model is presented and in [6] a constraint modeling-based approach is described. Although, these approaches contribute greatly, unfortunately, they are ungeneralizable, where previously unconsidered disciplines can be hardly integrated later. To solve this generalization issue, different approaches have been deployed. SysML [7] is one of these general-purpose approaches. For instance in [8], SysML was used to specify the central view-model of the mechatronics system. In [9], the system-level modeling with SysML was adopted to support mechatronic design. While in [10], SysML profiles were particularly applied to support the multi-view modeling approach.

From a requirements engineering point of view, various methods dealing with requirements analysis and traceability have been proposed. However, the mapping between requirements and system design model entities (i.e., components, properties) still rarely exists and, even if it did, it requires high synthesis and modification effort. Although SysML supports in modeling this mapping, its execution is still an open topic.

These previous works as well as others have contributed to the maturity of SysML. However, SysML does not have a formal semantic, is solely useful for project specific intentions, and lacks support of generalized execution. Extending our previous work [11, 12], while focusing on the system engineering level, we generalize the previous approaches by providing a mathematical formulation of the technical and economical aspects to support SysML execution and thus interoperability among the different design disciplines.

Artificial Intelligence (AI) methods have been proposed to aid the mechatronic design process. For instance, in [13] the design activity optimization was solved using a heuristic-based hybrid search algorithm and in [14] a maximum likelihood estimation method for determining the unknown design parameters based on given information was proposed. The main problems in existing approaches are twofold: (1) high effort in capturing the interdisciplinary information to be used in AI, and (2) problem specific design modeling and optimization, due to the adoption of parametric techniques. In our work we generalize the previous approaches, where we reduce the effort in providing the knowledge needed for AI through the proposed SysML model, and use non-parametric regression techniques to provide a problem independent design framework.

## 3 Background Preliminaries

This section presents background material needed to understand the remainder of the paper.

### 3.1 Mechatronic System Design

In theory, the VDI 2206 guideline [3], is one of the popular exemplifications of the mechatronic design process. It supports the creation of an interdisciplinary principal solution during the V-model's system design phase. Traditionally, during the initial design stages, the requirements are captured, categorized and analyzed. Therefore, modeling, analysis and simulations are the main activities performed in any mechatronic design methodology to assess a set of demanded requirements. Apart from the methodological aspects, different engineering tools are being employed and can be categorized into three types: (1) domain-specific tools (e.g., circuit design tools, software engineering tools, mechanical CAD tools), (2) domain-coupling tools (e.g., MATLAB, Modelica), and (3) all-in-one tools (e.g., Mechatronic Concept Designer).

In order to describe the disciplines' interdependencies between the tools, a document-based approach has been followed, such as Excel sheets, MS word, and/or PowerPoint files. This issue has been the reason of many project failures due to the lack of traceability and enactment of these interdisciplinary entities. Thus, this approach was over-thrown by the *Model-Based Systems Engineering* (MBSE) methodology. Here, models are used to represent such interdependencies and are intended to facilitate the design activities thus resulting in better communication, system design integration and system reusability [1].

### 3.2 Systems Modeling Language (SysML)

SysML is a "general-purpose graphical modeling language" [7]. It is developed as a software engineering extension of a customized subset of the *Unified Modeling Language* (UML) with the goal of being applied for systems' engineering applications. SysML captures the multidisciplinary knowledge by providing various diagrams: block definition, internal block, parametric and package diagrams to present the *structure* of the system. It further delivers activity, sequence, state machine and use case diagrams to describe the *behavior* of the product. Finally, with its major contribution, it allows for modeling the *requirements* of a system with the aid of its requirements' diagrams. It also integrates the previous three aspects (i.e., structure, behavior, and requirements) through allocations across their corresponding elements. SysML further offers a profile mechanism, where a profile is formed from a set of stereotypes of its elements. These stereotypes extend the syntax of SysML allowing it to be more applicable in concrete applications.

### 3.3 Gaussian Processes

Gaussian Processes (GPs) are a form of non-parametric regression techniques. Following the notation of [15], given a data set  $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^m$  where  $\mathbf{x} \in \mathbb{R}^d$  is the input vector,  $y \in \mathbb{R}$  the output vector and  $m$  is the number of available data points when a function is sampled according to a GP, we write,  $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ , where  $m(\mathbf{x})$  is the mean function and  $k(\mathbf{x}, \mathbf{x}')$  the covariance function, fully specifying a GP. Learning in a GP setting involves maximizing the marginal likelihood of Equation 1.

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T (\mathbf{K} + \sigma_n^2\mathbf{I})^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{K} + \sigma_n^2\mathbf{I}| - \frac{n}{2} \log 2\pi, \quad (1)$$

where  $\mathbf{y} \in \mathbb{R}^{m \times 1}$  is the vector of all collected outputs,  $\mathbf{X} \in \mathbb{R}^{m \times d}$  is the matrix of the data set inputs, and  $\mathbf{K} \in \mathbb{R}^{m \times m}$  is the covariance matrix with  $|\cdot|$  representing the determinant. GPs automatically avoid overfitting due to the presence of the second term in Equation 1 (i.e.,  $\frac{1}{2} \log |\mathbf{K} + \sigma_n^2\mathbf{I}|$ ). Due to space constraints we refer the interested reader to [15] for a thorough discussion of the topic.

## 4 The Need for a Unified Language and Adaptation

We adopt the V-model suggested by the VDI 2206 guideline [3], shown in Figure 1, as a macro-cycle consisting of requirements analysis, system design, domain-specific design and system integration phases that end with the product disposal. Despite the V-model's support for modeling and model analysis, the whole process is currently a theoretical construct without tool support. In addition, different gaps and short comings exist among the employed models as shown in Figure 1. These gaps affect the traceability and impede in updating the actuality of the different entities across the phases. Previous experience [12] has shown how the application of SysML in documenting such interdisciplinary relationships in a system model can glue these gaps.

These gaps could be traced back to problems in communication and integration among the different disciplines due to the lack of an efficient system model. In addition, another problem is the lack of adaptability and generalizability in the design process.

To solve these problems, we make use of SysML, as a common modeling language, to model three aspects of the system's design. Namely, we use the requirements (req), block definition (bdd), and parametric (par) diagrams to model the system's requirements, structure, and constraints respectively. In the sequel of making the design process adaptable to changing requirements and/or priorities as well as to support a multi-alternative design platform, we make use of GPs and optimization. In the following, the technical details will be further explained.

## 5 AI Support for Mechatronic System Design

In this section we will detail the proposed framework, to: (1) capture the interdisciplinary knowledge among the different involved domains, (2) provide the mathematical formulation of the requirements satisfaction problem, and (3) reflect upon the GP approximation used in our platform.

### 5.1 SysDICE Overall Framework

Figure 2 presents a high level scheme of the proposed framework. We categorize the human factors involved into (1) Discipline and (2) System engineers. For the first group, a discipline-specific information can be represented in SysML while assuring that the SysML detail level is restricted to only the amount of information needed for achieving a cross-discipline mapping. For the second category, system engineers, can model system requirements, the abstract conceptual solution and manage the system model using SysML. They are able to evaluate the system design model through MATLAB which is running in the background to provide a solver for SysML.

Furthermore, Figure 2 indicates three types of activities (i.e., requirements, structure, and constraints modeling) essential in any system design phase. Each of these activities results in a (set) of SysML diagrams. These diagrams provide a multidisciplinary model split into three fundamental levels: (1) the system's requirements with their desired numerical values and weighted priorities (e.g., total weight of 2 Kg with 70% priority), (2) the hierarchy of the components together with their respective parameters (i.e., components can be interdisciplinary, mechatronics, such as a motor with motor board controller or discipline-specific such as chassis as mechanical, electronic board as electrical or pure software code), and (3) the interrelationships between disciplines through the constraints with their corresponding input and output properties (e.g., power consumption, operational time, total price).

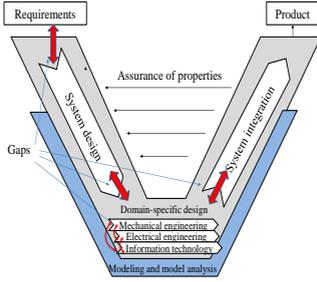


Figure 1: The V-Model as a macro-cycle describes the generic procedure for designing mechatronic systems [3].

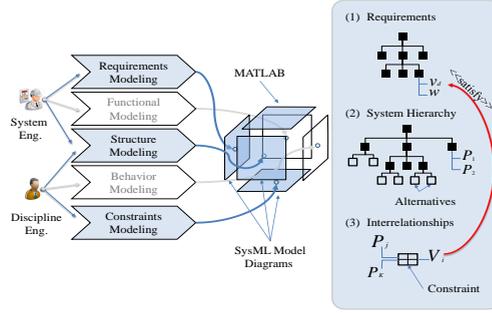


Figure 2: The SysDICE framework: Five modeling activities result in a set of SysML diagrams (The functional and behavior modeling are future work).

In this manner a system model holds all the necessary interdisciplinary relations, constraint information, different component alternatives as well as requirements values and priorities in one unified model. This unified SysML model is then converted into MATLAB for evaluating different configurations of requirements' values and priorities. This evaluation is conducted with the goal of attaining the best component alternative to suit the customers' objectives. For that, a theoretical system design model is presented in Section 5.2 and a mathematical optimization problem is formulated and solved as described in Section 5.3.

## 5.2 System Design Model

During early design stages a set of requirements spanned over the various domains is provided. In our framework, each of these requirements is modeled using the  $\ll requirement \gg$  block within the req diagram. To be fully able to specify a numerical design requirement, we extend the existing SysML requirement block by stereotyping it to include its value,  $v_d$  and its corresponding priority,  $w$ . We further consider the hierarchy of the requirements using the *containment* relationship for the traceability.

In industry, after the design requirements have been settled, system engineers commence to analyze the type of system satisfying such requirements. At this stage, the system evolves from a black box to detailed subsystems reaching the component levels. Following a similar trend, our framework then decomposes the system into its constituent subsystems and their corresponding components. This is achieved through the SysML  $\ll block \gg$  element and the *composition* association within the bdd diagram. Each component of the system has various alternatives which are modeled with a stereotyped  $\ll block \gg$  in order to represent their uniqueness in a possible conceptual design solution. They are specified by their corresponding properties such as the weight, the price, the power consumption and so forth. The relations between these properties are modeled using the  $\ll constraintProperty \gg$  within the par diagram.

Additionally, the system design model is generated in an iterative and evolutionary manner with each of the three activities. At the stage where the model is fully specified from the requirements down to the properties level, the goal then is to find the optimal alternative combination that best suits the prioritized, and possibly conflicting requirements. Therefore, the stereotyped requirements with corresponding values (i.e.,  $\mathbf{v}_d$  and  $\mathbf{w}$ ) as well as all other blocks with their respective properties are transformed to MATLAB. The constraint properties with their MATLAB-based equations are transformed into MATLAB functions. In the next section we provide the mathematical formalization of the weighted requirement satisfaction problem.

## 5.3 Mathematical Formulation

Given a set of  $k$  requirements, we define  $\mathbf{v}_d = [v_d^{(1)}, \dots, v_d^{(k)}]^T \in \mathbb{R}^{k \times 1}$  to represent the different desired values of each of the requirements, and  $\mathbf{W}_{k,k} = \text{diag}(\mathbf{w})$  to be the diagonal matrix representing the priorities of each of these requirements. We further define  $\mathbf{v} = [v_1, \dots, v_k]$ , to represent the output of the constraint equations which relate a set of priorities as its inputs.

We assume that these values are noisy<sup>1</sup>, with a gaussian noise, and that the requirements are weighted in each of the  $k$  directions according to their priorities. Therefore, the likelihood for a desired value to occur is

<sup>1</sup>We assume that the combination and or values of the properties are not exact and rather noisy.

defined by,

$$p(v_d^{(i)} | v^{(i)}; \sigma^2, w^{(i)}) = \prod_{i=1}^k \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} w_{i,i} (v_d^{(i)} - v^{(i)})^2\right) \quad (2)$$

Maximizing the natural logarithm of Equation 2, leads to the following minimization problem,

$$\min_{\mathbf{v}} \frac{1}{2} [\mathbf{v} - \mathbf{v}_d]^T \mathbf{W} [\mathbf{v} - \mathbf{v}_d] \quad (3)$$

Equation 3 represents the weighted requirement satisfaction problem. In other words, the solution of the minimization problem is seeking the optimal value,  $\mathbf{v}^*$ , so as to minimize the error with respect to the desired combination weighted by the priorities (i.e.,  $\mathbf{v}^* = \arg \min_{\mathbf{v}} \frac{1}{2} [\mathbf{v} - \mathbf{v}_d]^T \mathbf{W} [\mathbf{v} - \mathbf{v}_d]$ ).

To approximate the values of the corresponding combination of the properties, we resort to GPs. The reasons for our choice is threefold: (1) the constraint equations are complex and thus require non-parametric functional approximators, (2) the lack of available training data which imposes good generalization properties of the used approximators, and (3) the need for a problem independent framework.

The approximated functions are then substituted in Equation 3, to generate a new minimization problem defined by the following cost function,

$$\min_P J(P) = \frac{1}{2} \sum_{i=1}^k w_{i,i} (\mathcal{GP}_i(P) - v_d^{(k)})^2, \quad (4)$$

where  $P = p_1 \otimes p_2 \cdots \otimes p_N$ , with  $N$  being the number of components, and  $J$  representing the cost function.

To minimize Equation 4, we need to compute the derivatives with respect to the input. Here we approximate the derivate of a GP using first order approximation and then use conjugate gradient descent for the optimization. The output is  $P^*$  that satisfies the set combination of the prioritized requirements (i.e.,  $\arg \min_P \sum_{i=1}^k w_{i,i} (\mathcal{GP}_i(P) - v_d^{(k)})^2$ ).

## 6 Experiments and Results

In this section we explain two different design experiments that were conducted.

### 6.1 Experiment One: Two Wheel Differential Drive Robot

The first experiment illustrates the design of a two wheel differential drive robot. The *e-puck*<sup>2</sup>, top-right of Figure 3, is an example of such robots. Next we describe the application of our proposed framework in: (1) modeling the robot using SysML and (2) using the mathematical formulation and GPs to find the optimal combination of component alternatives to satisfy different requirements' configurations.

#### 6.1.1 SysML Model Generation

During the initial stages of the robot's system design phase, system engineers identified robot's requirements as well as the possible conceptual solutions and discipline engineers detailed the solution concepts with their domain-specific information and the possible alternatives. Conclusively, a system model of the robot with SysML was achieved based on these information. SysML modeling was done using the open source tool TOPCASED-SysML [16].

Figure 3 shows the three types of SysML diagrams: req, bdd, par diagrams used to model the required information of the mobile robot. The top-left of Figure 3 shows a part of the main design requirements: the *TotalWeight*, the *TotalPrice*, the *MaximumTranslationalVelocity*, and the *OperationTime*. Each is stereotyped as "REQ" to allow for the addition of the requirements' properties (i.e.,  $v_d$  and  $w$ ). Similarly all other requirements were modeled. Each REQ must be satisfied by a value of a design entity (i.e. component, property or even a system). Therefore, the  $\ll satisfy \gg$  association was used to represent which design entity satisfy which requirement.

The robot components are modeled using bdds. We model the components hierarchy, using the SysML  $\ll composition \gg$  relationship. Figure 3 details modeling these components. Each component of the system is described using its own block that holds certain properties typically needed by the engineer during the design phase. In our example the robot consisted of 7 different components, each having its own alternatives. These alternatives are modeled with blocks that are stereotyped as "ALT" so to indicate the multi-alternatives for each component during the transformation (e.g., Motor1Type1, Motor1Type2).

<sup>2</sup>e-puck: <http://www.e-puck.org/>

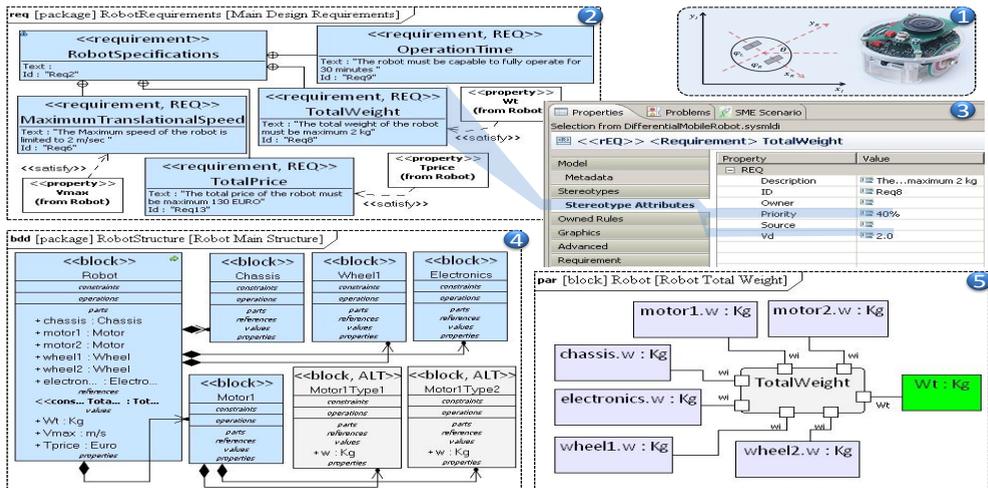


Figure 3: Robot SysML diagrams: (1) e.g. the *e-puck*, (2) requirements with (3) *TotalWeight* properties, (4) bdd for components' structure and alternatives and (5) *par* for the *TotalWeight* constraint property.

Various *par* diagrams were used to model the mathematical equations between the component properties. Each equation is represented with a *constraintProperty* with its own input and output properties. For instance, the constraint “*TotalWeight*” is used in the *par*, Figure 3, to relate all the components' weight properties (component.w) thus indicating the value of the actual total weight of the robot  $W_t$ . Here the *TotalWeight REQ* is satisfied by this property  $W_t$  that indicates the actual value  $v$ . The kinematical, dynamical as well as other related equations, such as the total power consumption, the total price, and the operational time have been also modeled similarly with other *par* diagrams. At this stage a SysML model incorporating all the disciplines is generated after several iterations. Therefore, the necessary information for system engineers is ready and the communication burden is solved.

**Conclusion I:** SysML can serve in bridging the communication problem.

**6.1.2 SysML Model Evaluation**

To better evaluate the framework, we have conducted various experiments with different priorities and desired values of the requirements. The system was provided with different alternatives having different properties, such as, the mass, the price and so forth as described above. The algorithm was provided with different  $v_d$ 's and  $w$ 's. After the GPs were approximated, conjugate gradient descent was applied to find the optimal alternative suiting the requirements. The values corresponding to the properties of the determined alternative could be seen in the appendix<sup>3</sup>. Figure 4 shows the results of providing different values and priorities. The three axis of the graph represent the components, properties and the alternatives respectively. The different planes are the optimal alternatives resulting from different requirements' values and priorities. Each of these priorities and/or properties change represents a different design focus. For instance, in the first plane (1<sup>st</sup> alternative) the focus was more towards having a high velocity robot (i.e., 2 m/s) with high operational time (i.e., 1 hour), where both requirements were given a priority of 90%. The second plane (4<sup>th</sup> alternative) represents a moderate robot while the third

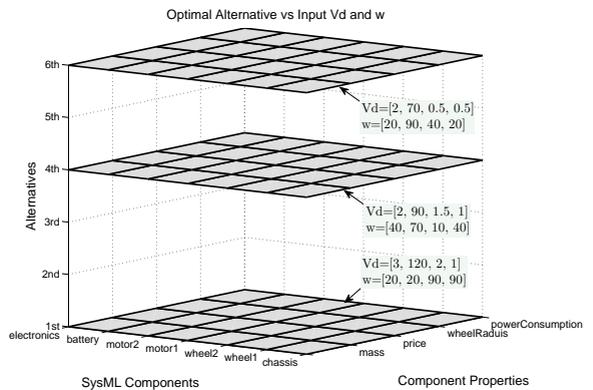


Figure 4: Results on three different design experiments. Each plane represents the optimal alternative for the corresponding requirement priorities and values.

<sup>3</sup>Appendix published online at: <https://dl.dropbox.com/u/2689877/bnaic2012IntelligentDesignAppendix.pdf>

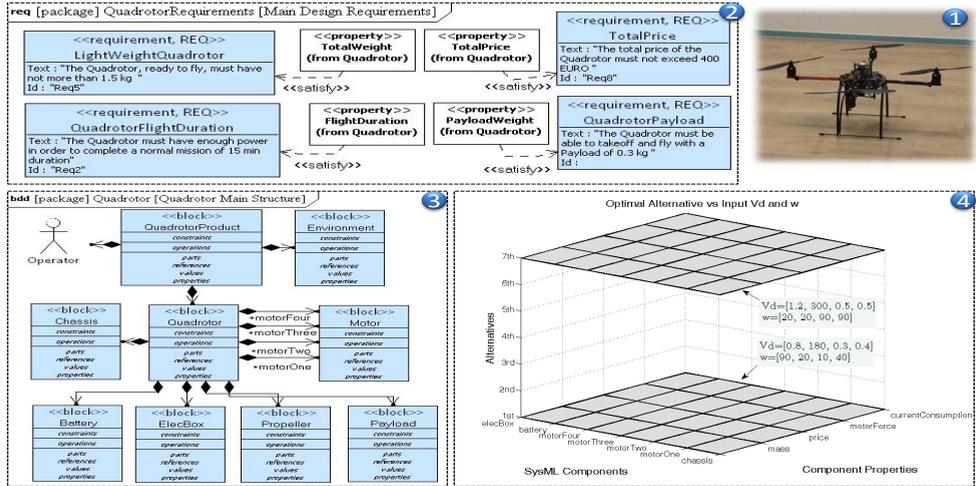


Figure 5: (1) the quad-rotor system present at the Swarm-Lab in the Department of Knowledge Engineering, Maastricht, The Netherlands, (2) a req diagram showing a part of the main design requirements satisfied by their respective properties, (3) a bdd presenting the quad-rotor components structure and (4) results of two different design focus requirements.

(6<sup>th</sup> alternative) correspond towards having a cheap price robot of 70 € with a high priority (i.e., 90%). It becomes obvious from Figure 4 that the platform was capable of capturing different optimal alternatives suiting different design focuses and requirements and thus being adaptable and generalizable.

**Conclusion II:** The proposed framework is capable of attaining the optimal combination to suit a set of prioritized requirements.

**Conclusion III:** The proposed framework is capable of attaining different optimal alternative solutions to different design focuses and thus being adaptable.

## 6.2 Experiment Two: Quad-rotor

To better assess the design and the evaluation process, we have conducted a second more complex design task. In this experiment a quad-rotor unmanned aerial vehicle, shown at the top-right of Figure 5 was designed. The quad-rotor is a system consisting of four rotors in a square connection. The dynamics of the system are represented by a 12-dimensional state-vector and the actions are different torques delivered by the motors. In this task we had more constraints taken into account as well as more alternatives.

### 6.2.1 SysML Model Generation

The SysML model generation phase was generated similarly to the one described in the previous experiment. Here the system had to satisfy four requirements. Namely, *LightWeightQuadrotor*, *TotalPrice*, *QuadrotorFlightDuration* and *QuadrotorPayload*, shown in Figure 5. Further, the components as well as the constraints were modeled using the corresponding diagrams similarly to the last experiment.

### 6.2.2 SysML Model Evaluation

We have also conducted various experiments with different priorities and desired values of the requirements. The system was provided with different alternatives having different properties. The algorithm was provided with different  $v_d$ 's and  $w$ 's. After the GPs were approximated, conjugate gradient descent was applied to find the optimal alternative suiting the requirements as described in Section 5.3. The bottom-right side of Figure 5 shows the results of providing different values and priorities for the requirements in two experiments (the properties' values corresponding to all possible alternatives could be seen in the appendix<sup>3</sup>).

Here also, the three axis of the graph represent the components, properties and the alternatives respectively. The different planes are the optimal alternatives resulting from different requirements' values and priorities. Each of these priorities and/or properties change represent a different design focus. It is clear from the values that the first plane (1<sup>st</sup> alternative) correspond to a low-weight quad-rotor (0.8 kg) with priority of 90% while the second plane (7<sup>th</sup> alternative) is a result of a quad-rotor with high flight duration (0.5 hour) and high payload (0.5 Kg) with both a 90% priority. Similar conclusions could be drawn from this

experiment, where the proposed framework bridges the communication gap, can attain optimal alternative combination and is adaptable.

## 7 Conclusions and Future Work

In this paper we presented a SysML-based approach in order to support the design of mechatronic systems. By leveraging SysML, the platform was capable of incorporating the interdisciplinary interrelations that go with and complicate the design process. The framework was split into three fundamental levels that are typically used in the design process. It further makes use of Gaussian Processes in order to find a functional mapping at the system-design level. These are then used to solve for the best alternative that optimally suits a set of requirements. Experiments conducted on the design of two systems, show the accessibility and adaptability of the approach, whereby the framework was capable of bridging the system engineering level communication problems, attaining optimal alternatives to a set of requirements, and producing adaptable solutions to various design focuses.

In future work, we aim to extend the actual system model with the interfaces across components in order to restrict the space of alternative exploration to suit the requirements. On a higher level, other discipline-specific information, the functional and the behavior aspects, will be incorporated in the existing system model. Moreover, we are in a sequel of using transfer learning to adapt already learned behaviors in similar designs of similar systems.

## References

- [1] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*. No. ISBN: 978-0123743794, Elsevier, Morgan Kaufmann OMG Press, 2008.
- [2] G. Pahl, W. Beitz, J. Fledhusen, and K.-H. Grote, *Engineering Design A Systematic Approach*. Springer, third edition ed., 2007.
- [3] *VDI 2206 Design methodology for mechatronic systems*. Beuth Verlag GmbH, June 2004.
- [4] F. P. Stappers, L. J. Somers, and M. A. Reniers, "Multidisciplinary Modeling - Current status and expectations in the Dutch TWINS consortium," in *ICSSEA*, 2008.
- [5] A. A. A. Cabrera, M. S. Erden, M. J. Foeken, and T. Tomiyama, "High Level Model Integration for Design of Mechatronic Systems," *IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, October 2008.
- [6] K. Chen, J. Bankston, J. H. Panchal, and D. Schaefer, *A Framework for Integrated Design of Mechatronic Systems*, ch. 2, pp. 37–70. Springer, 2009.
- [7] "Object Management Group (OMG) Systems Modeling Language (OMG SysML™), available at <http://www.omgsysml.org> .," November 2008.
- [8] K. Thramboulidis, "The 3+1 SysML View-Model in Model Integrated Mechatronics," *Journal of Software Engineering and Applications (JSEA)*, vol. 3, no. 2, pp. 109–118, 2010.
- [9] A. Qamar, J. Wikander, and C. During, "Designing Mechatronic Systems: A Model-Integration Approach," in *Proceedings of the 18th International Conference on Engineering Design (ICED11)*, vol. 4, pp. 145–156, 2011.
- [10] A. A. Shah, D. Schaefer, and C. J. Paredis, "Enabling Multi-View Modeling With SysML Profiles and Model Transformations," in *International Conference on Product Lifecycle Management*, p. 10, Inderscience Enterprises Ltd, 2009.
- [11] M. Chami, H. Seemller, and H. Voos, "A SysML-based Integration Framework for the Engineering of Mechatronic Systems," *IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, IEEE, 2010.
- [12] R. Stetter, H. Seemller, M. Chami, and H. Voos, "Interdisciplinary System Model for Agent-Supported Mechatronic Design," in *Proceedings of the 18th International Conference on Engineering Design (ICED11)*, Vol. 4, 2011.
- [13] O. Mouelhi, P. Couturier, and T. Redarce, "An Artificial Intelligence Approach for the Multicriteria Optimization in Mechatronic Products Design," in *Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation*, pp. 1731–1736, 2009.
- [14] X. Xu, L. Fu, and S. Fang, "Research on Product Variant Design with Uncertainty Information," in *Proceedings of the 7th World Congress on Intelligent Control and Automation*, (Chongqing, China), June 25 - 27 2008.
- [15] C. E. Rasmussen and C. K. I. Williams, "Gaussian processes for machine learning," MIT Press, 2006.
- [16] "TOPCASED-SysML Modeling Framework Open Source Project, <http://www.topcased.org> .."

# Transfer Learning for Bilateral Multi-Issue Negotiation

Siqi Chen <sup>a</sup>      Haitham Bou Ammar <sup>a</sup>      Karl Tuyls <sup>a</sup>      Gerhard Weiss <sup>a</sup>

<sup>a</sup> *Department of Knowledge of Engineering, Maastricht University, The Netherlands*

## Abstract

This paper proposes a novel strategy named Transfer between Negotiation Tasks (**TNT**) for automated bilateral negotiation with multiple issues. **TNT** is able to probabilistically transfer between different negotiation tasks in order to bias the target agent’s learning behavior towards improved performance without unrealistic assumptions. We analyze the performance of our strategy and show that it substantially outperforms a powerful negotiation strategy across a variety of negotiation scenarios.

## 1 Introduction

Automated negotiation has been achieving steadily growing attention as a coordination mechanism for interaction among computational autonomous agents which are in a consumer-provider or buyer-seller relationship and thus typically have different interests over possible joint contracts. Automated negotiation would come in many shapes and forms, for instance, sequential versus concurrent negotiation (i.e., multiple negotiations occur one after the other or at the same time), bilateral versus multi-lateral negotiations (i.e., an agent negotiates with a single other agent or multiple agents are involved in a single negotiation at the same time), and single-issue versus multi-issue negotiation (i.e., a single or several issues are subject of a negotiation among agents) and so on.

Given the pervasive nature of automated negotiation, negotiating agents are required to obtain a high level of self-determination, whereby they decide for themselves what, when and under what conditions their actions should be performed to reach a satisfactory agreement. This objective is however difficult to achieve mainly due to the lack of sufficient knowledge about opponents. To address this problem, existing work concentrates on opponent modeling. For instance, Saha et al. [11] applies Chebychev polynomials to estimate the chance that the negotiation partner accepts an offer in repeated single-issue negotiations on the same domain. In this setting the opponent’s response can only be acceptance or rejection to a certain offer. In [2], Brzostowski et al. investigate the online prediction of future counter-offers on the basis of previous negotiation history by using differentials, assuming that the opponent’s strategy is known to be based on a mix of time- and behavior-dependent one. In [3] an artificial neural network (ANN) is used to compete against human negotiators in a specific domain, its training however requires a very large database of previous encounters. It is clear that the previously done work assumes certain additional structural assumptions to guarantee the effectiveness of the proposed approach. Apart from the structural assumptions, learning in different negotiation settings within the same domain is slow due to the lack of enough information about the opponent.

Transfer Learning (TL) [1, 7], is one approach that aims at improving learning times and/or performance by leveraging already acquired knowledge in similar tasks. In this work we propose the first, to the best of our knowledge, transfer learning negotiation framework titled “Transfer between Negotiation Tasks (**TNT**)”. **TNT** is capable of probabilistically transferring between different negotiation tasks in order to bias the target agent’s learning behavior towards better performance. **TNT** also aims at solving the structural assumption problem by relaxing the previous work, where it doesn’t introduce any additional

assumptions on the structure of the opponent’s model and/or its behavior. It rather makes use of nonparametric regression techniques namely, Gaussian Processes (GPs) to learn the opponent’s model. In this sense, TNT advances the state-of-the-art bilateral multi-issue negotiation by contributing in: (1) proposing a first-of-its-kind transfer opponent modeling framework, (2) outperforming tough target negotiation strategies using the probabilistic strategy transfer mixture, and (3) providing a problem independent negotiation transfer scheme, where the type of function approximators don’t restrict the framework from any direction.

The remainder of this paper is structured as follows. Section 2 describes the background preliminaries of this work. Section 3 presents the TNT strategy. The performance evaluation is given in section 4. Section 5 discusses experimental results. Lastly, Section 6 concludes and identifies some important research lines induced by the work.

## 2 Background Preliminaries

### 2.1 Bilateral Negotiation

We adopt a basic bilateral multi-issue negotiation model which is widely used in the agents field (e.g., [5]) and the negotiation protocol we use is based on a variant of the alternating offers protocol proposed in [8]. Let  $I = \{a, b\}$  be a pair of negotiating agents, and let  $i$  ( $i \in I$ ) represent a specific agent. The goal of  $a$  and  $b$  is to establish a contract for a product or service. Thereby a contract consists of a vector of values, each assigned to a particular issue such as price, quality and delivery time. Agents  $a$  and  $b$  act in conflictive roles. To make this precise, let  $J$  be the set of issues under negotiation and  $j$  ( $j \in \{1, \dots, n\}$ ) be a particular issue. Each agent has a lowest expectation for the outcome of a negotiation; this expectation is called reserved utility  $u_{res}$ . During negotiation, each issue  $j$  gets assigned a value  $O_j$ . The tuple  $O = (O_1, \dots, O_n)$  is called a contract. A contract is said to be established if both agents agree on it.

Following Rubinstein’s alternating bargaining model [10], each agent makes, in turn, an offer in form of a contract proposal. An agent receiving an offer needs to decide whether to accept or reject it and to propose a counter-offer. In the case an agent’s deadline is reached, it has to withdraw from the negotiation. The agents decide as follows. Each agent has a weight vector (also called importance vector or preference vector) over the issues, representing the relative importance it assigns to each of them. The weight vector of agent  $i$  is written as  $w^i = (w_1^i, \dots, w_n^i)$ , where  $w_j^i$  ( $j \in \{1, \dots, n\}$ ) is the weight (or preference) which agent  $i$  assigns to issue  $j$ . The weights of an agent are normalized (i.e.,  $\sum_{j=1}^n (w_j^i) = 1$  for agent  $i$ ). The utility of an offer for agent  $i$  is obtained by the utility function, defined as:

$$U^i(O) = \sum_{j=1}^n (w_j^i \cdot V_j^i(O_j)) \quad (1)$$

where  $w_j^i$  and  $O$  are as defined above and  $V_j^i$  is the evaluation function for  $i$ , mapping every possible value of issue  $j$  (i.e.,  $O_j$ ) to a real number.

After receiving an offer from the opponent,  $O_{opp}$ , an agent decides on acceptance or rejection according to its interpretation  $I(t, O_{opp})$  of the current negotiation situation. For instance, this decision can be made depending on a certain threshold or can be based on utility differences. Negotiation continues until one of the negotiating agents accepts or withdraws due to timeout<sup>1</sup>.

### 2.2 Gaussian Processes

Gaussian Processes (GPs) are a form of nonparametric regression techniques that perform inference directly in the functional space. In other words, GPs define probability distributions over functions. Concretely, given a data set  $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^m$  where  $\mathbf{x} \in \mathbb{R}^d$  is the input vector,  $y \in \mathbb{R}$  the output vector and  $m$  is the number of available data points, when a function is sampled from a GP, we write:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')),$$

<sup>1</sup>If the agents know each other’s utility functions, they can compute the Pareto-optimal contract [8]. However, a negotiator will not make this information available to its opponent in general.

where  $m(\mathbf{x})$  the mean function, and  $k(\mathbf{x}, \mathbf{x}')$  the covariance function that both fully specify a GP.

Learning in a GP setting involves maximizing the marginal likelihood given by Equation 2.

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T (\mathbf{K} + \sigma_n^2\mathbf{I})^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{K} + \sigma_n^2\mathbf{I}| - \frac{n}{2} \log 2\pi, \quad (2)$$

where  $\mathbf{y} \in \mathbb{R}^{m \times 1}$  is the vector of all collected outputs,  $\mathbf{X} \in \mathbb{R}^{m \times d}$  is the matrix of the input data set, and  $\mathbf{K} \in \mathbb{R}^{m \times m}$  is the covariance matrix with  $|\cdot|$  representing the determinant. It is interesting to see that GPs automatically avoid over-fitting. Maximizing Equation 2 is attained according to conjugate gradient descent; refer to [9] for a thorough discussion of the topic.

### 2.3 Transfer Learning

Learning in new negotiation tasks is expensive due to either the lack of enough information, or to the complexity of the task. Transfer Learning (TL) is a technique that leverages the usage of information in a source task to aid learning in a target. In TL settings, there typically exists a source and a target task. The agent is assumed to have learned a model of the source task and is now faced by a new target task where little or no information is present. Using the source task knowledge, the target task agent can bias its learning to increase and/or improve the learning speeds, the quality of the learned behavior and/or the overall performance. In this work we focus on transfer learning in supervised learning tasks, where we transfer source task models to aid the target task agent in learning against a new opponent strategy. Our main focus here is to transfer opponent models in the same negotiation domains. In other words, we re-use already learned utility modeling results from a source opponent to aid in learning the strategies against a different opponent in a target negotiation task. There are a lot of directions for this knowledge re-use, such as [7]. In this work we propose a new transfer in negotiation scheme where the source task knowledge is probabilistically re-used in a target negotiation one. The technicalities of the proposed framework are explained and discussed next.

## 3 Transfer between Negotiation Tasks (TNT)

### 3.1 Learning in the Source Task

The source negotiation task starts by the opponent agent presenting an offer describing values for the different negotiation issues. Our utility is calculated according to the proposed opponent's offer, which is either accepted or rejected. If the offer is accepted the negotiation session ends. On the other hand, if the offer is rejected our agent proposes a counter-offer to the opponent. Here the opponent can decide, according to his own utility function, whether to accept or reject our counter-offer. The opponent utility function is unknown to our agent rather it tries to learn it incrementally over time. This opponent utility is indirectly modeled from the utilities that our agent attains through the opponents counteroffers. To better clarify, once the opponent agent decides to propose a counteroffer to ours, we calculate the utility we get from its counteroffer and add it to the data set  $\mathcal{D}_1 = \{t^{(i)}, u^{(i)}\}_{i=1}^{t_{\max}}$ . It is worth noting that this is not a one shot learning. In other words, the data set grows dynamically as the negotiation session continues where the model is trained again with the addition of the new attained data points<sup>2</sup>. Once the  $\mathcal{D}_1$  is collected the nonparametric GP approximators are used to learn the opponent utilities.

The negotiation starts by the opponent proposing an offer that is accepted or reject by our agent lines 1–4 of Algorithm 1. If our agent accepts the proposed offer then the negotiation session ends, line 5. In case there was no agreement we propose a new offer that the opponent has to assess. We use the counteroffer produced by the opponent to approximate his utility indirectly using  $\mathcal{GP}_1$ , lines 7–15.

### 3.2 Knowledge Transfer and Target Task Learning

After the source agent had learned to negotiate against the source task opponent, it is now faced with a new opponent within the same negotiation domain. This target opponent differs from the source one by having

<sup>2</sup>In this work we split the negotiation session in intervals of 3 sec.

---

**Algorithm 1** Source Task Utility Learner

---

**Require:** Two negotiation agents, maximum time interval  $t_{\max}$ 

```

1: while  $t < t_{\max}$  do
2:   Opponent proposes offer
3:   Calculate utility
4:   if Accept then
5:     agreement reached
6:   else
7:     Propose newoffer
8:     if Opponent Accept then
9:       agreement reached
10:    else
11:      Opponent proposes counteroffer
12:      Collect time and utility and add to  $\mathcal{D}_1$ 
13:      Use GPs to approximate the opponents utility
14:      Use approximated opponents utility function to present an offer
15: return Opponent's utility model  $\mathcal{GP}_1$ 

```

---

a different negotiation strategy that could be more or less powerful. The idea is that the model learned against the source opponent will help in exploiting and learning against the target one. We make use of the source task's GP model by probabilistically proposing a combination of transfer and tough offers to the target opponent. In this context, tough refers to offers randomly produced adhering to a certain utility threshold value. To better clarify, the predicted outputs of the source task approximation model (i.e.,  $\mathcal{GP}_1$ ), are probabilistically used in combination with a tough offering strategy to aid the target task learner in approximating the target task opponent's model. This transfer approach is better described in Algorithm 2.

---

**Algorithm 2** Transfer between Negotiation Tasks (TNT)

---

**Require:** Two negotiation agents, maximum time interval  $t_{\max}^{(2)}$ ,  $\mathcal{GP}_1$ ,  $\epsilon$ 

```

1: while  $t^{(2)} < t_{\max}^{(2)}$  do
2:   Opponent proposes offer
3:   Calculate utility
4:   if Accept then
5:     agreement reached
6:   else
7:     Propose newoffer according to,
      
$$u_2 = \begin{cases} \text{tough offer,} & \text{with } p(\epsilon) \\ \text{transfer offer} & \text{with } 1 - p(\epsilon) \end{cases}$$

8:     if Opponent Accept then
9:       agreement reached
10:    else
11:      Opponent proposes counteroffer
12:      Collect time and utility and add to  $\mathcal{D}_2$ 
13:      Use GPs to approximate the opponents utility
14:      Use approximated opponents utility function
15: return Opponent's utility model  $\mathcal{GP}_2$ 

```

---

Algorithm 2 requires a target negotiation task, a maximum time interval  $t_{\max}^{(2)}$  presenting the end of the target negotiation task, as well as the source task approximated model  $\mathcal{GP}_1$ . The negotiation procedure commences similarly to that in Algorithm 1, where the target opponent proposes an offer that our agent either accepts or rejects. In the acceptance case, the negotiation is terminated as an agreement has been reached, lines 1–3. If our agent decides to present a newoffer it makes use of: (1) a tough strategy and (2) a

transfer strategy, as shown in line 7 of Algorithm 2. This trade-off between a tough and a transfer strategy is probabilistic where the agent probabilistically decides which offer to propose according to what strategy. In case it decides to make use of the source task knowledge in order to explore the target opponents behavior, it uses the output of the approximated source task model (i.e.,  $\mathcal{GP}_1$ ) in order to propose an offer<sup>3</sup> according to,  $u_2 \sim \mathcal{GP}_1(\mathbf{m}_1(t^{(2)}), \mathbf{k}_1(t^{(2)}, t'^{(2)}))$ , where  $\mathbf{m}_1(t^{(2)})$  presents the already learned mean function of  $\mathcal{GP}_1$  in the source negotiation task evaluated at the target task's learning time, and  $\mathbf{k}_1(t^{(2)}, t'^{(2)})$  represents the already fitted source task covariance function also evaluated at the target task's time<sup>4</sup>.

Once these offers are proposed, the target opponent can either accept or reject. In the rejection case, it will propose a counteroffer, in which the utilities are collected from and added to the data set  $\mathcal{D}_2$ , line 8–12. Then a GP is used to approximate the target opponent's utility, as shown in lines 14 – 15 of Algorithm 2. In expectation this added knowledge is expected to improve the performance of our target agent in approximating the target's opponent utility. This view is solidified and shown applicable in eight experimental domains described next.

## 4 Experiments

The performance evaluation of TNT is done with GENIUS (General Environment for Negotiation with Intelligent multipurpose Usage Simulation [6]) which is also used as a competition platform for the international Automated Negotiating Agents Competition (ANAC). It allows to compare agents (representing different negotiation strategies) across a variety of application domains under real-time constraints, where the preference profiles of two negotiating agents are specified for the individual domains.

In the first set of experiments, we first carry out the source task, which is a negotiation session against a weak opponent according to the ANAC ranking. The source task is done separately in two domains, namely, *Amsterdam Party* and *Travel*. The knowledge gained from previous negotiations is then used by the **TNT-agent** (the implementation of TNT) to adjust/optimize its strategy in target tasks, where it competes with different opponents (i.e., the agents who use a weak or strong strategy, which is classified according to the performance in ANAC). For the second set of experiments, the knowledge comes from a bargaining process with a strong agent in the same two domains (i.e., *Amsterdam Party* and *Travel*).

In target tasks, we build a simple but strong negotiation strategy named **Tough** as the basic benchmark. **Tough** will propose random offers whose utilities are above a certain threshold  $\alpha$  over the negotiation course (in our experiments  $\alpha$  was set to 0.75.) The weak agent in this task is Agent.K2, who is in the fifth place at the ANAC2011, meanwhile the strong agent is **Hardheaded**, who was the champion of ANAC2011.

### 4.1 Weak Source Task

In this set of experiments, we choose **Iamhaggler2011** as the weak opponent and the **OMACagent** [4] as our agent. **OMACagent** finished in the third place at the ANAC2012, and is able to advance **Iamhaggler2011** (the third of ANAC2011) by a significant margin. The utilities of counter-offers from **Iamhaggler2011** are used by the Gaussian process to approximate the received opponent utility points, and the resulting predicted values are transferred to the target task.

Figure 1 shows the performance between transfer and no transfer strategies in the domain *Amsterdam Party* playing against Agent.2K. The  $x$ -axis represents time percentage elapsed in negotiation and  $y$ -axis presents the utility. During the early phase of bargaining, no significant difference can be observed. **TNT-agent** however outperforms its counterpart on the late stage of negotiation. When the negotiations happen in *Travel* given in Figure 2, the **TNT-agent**, like the **Tough**, does not obtain obvious concession from Agent.2K on the early stage, meanwhile the gap between the two strategies becomes larger and larger as the negotiation is approaching the end.

The performance difference when negotiating with the hard opponent, *Hardheaded*, is demonstrated in Figure 3 and 4. It is very similar to what we observed before, that is, there is no obvious difference between these two strategies in first 80% of the negotiation. But, the **TNT-agent** again performs much better than the **Tough** in the remaining time. Another interesting observation is that in Figure 4 the hard opponent

<sup>3</sup>Please note that as the offers' utilities change between the source and the target task, we force the target task agent to adhere to a certain upper-bound once transferring.

<sup>4</sup>Please note that the GP prediction produces a multi-dimensional gaussian distribution and therefore a mean vector and a covariance matrix.

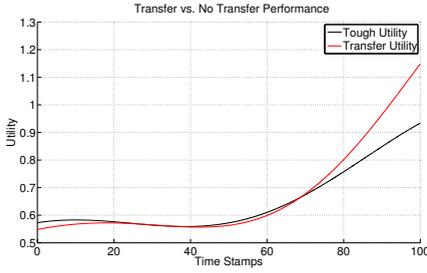


Figure 1: Negotiation with an easy opponent (Agent\_2K) in *Amsterdam Party* domain based on easy source task.

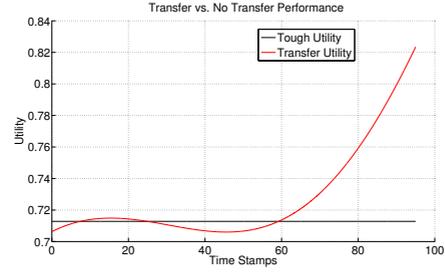


Figure 2: Negotiation with an easy opponent (Agent\_2K) in *Travel* domain based on easy source task.

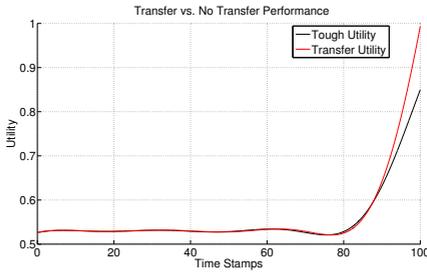


Figure 3: Negotiation with a hard opponent (Head-headed) in *Amsterdam Party* domain based on easy source task.

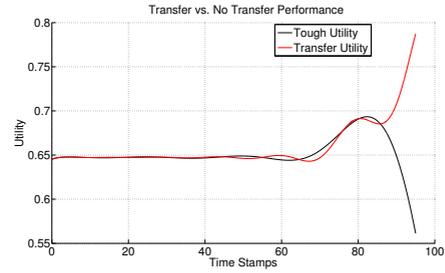


Figure 4: Negotiation with a hard opponent (Head-headed) in *Travel* domain based on easy source transferred to the target task.

made adverse concession with the purpose of achieving better agreement for itself when bargaining with Tough. TNT-agent, however, can avoid the negative behavior through combing the learned knowledge with its strategy.

**Conclusion I:** Transfer from a weak source negotiation task is capable of outperforming both an easy and hard target task opponent in expectation.

## 4.2 Hard Source Task

In the second set of experiments, the opponent in the source task switches to a strong one (i.e., Gahboninho, the second place of ANAC2011) while we keep using OMACagent as our agent. This is since Gahboninho is very strong in these domains and is able to achieve a very close score to OMACagent. According to Figure 5 and 6, when negotiating with the weak opponent (Agent\_2K) in the domains, TNT-agent outperforms the no transfer learning approach except for a short period in *Travel* (i.e., 20 time stamps). The transfer strategy outperforms the Tough one in the later phase of negotiation.

Then, in experiments where our agents play against the hard opponent using knowledge from hard source task, TNT-agent still performs well. For the two scenarios given by Figure 7 and 8, TNT-agent shows similar behavior. More specifically, its performance is close to the Tough for first 70%, and advances increasingly larger on the closing stage.

**Conclusion II:** Transfer from a hard source negotiation task is capable of outperforming an easy and hard target task opponents in expectation.

## 5 Discussions

In this section we will discuss the applicability scope of TNT as well provide an intuition on why transfer learning works in such a setting.

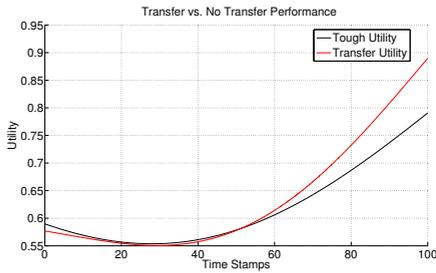


Figure 5: Negotiation with an easy opponent (Agent\_2K) in *Amsterdam Party* domain based on hard source task.

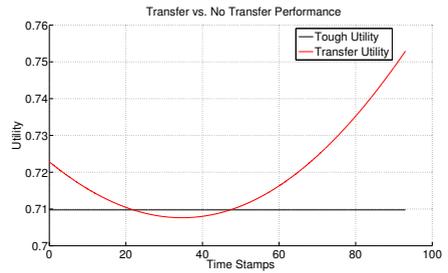


Figure 6: Negotiation with an easy opponent (Agent\_2K) in *Travel* domain based on hard source task.

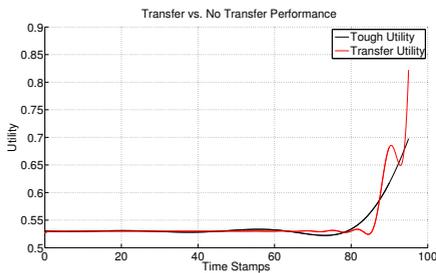


Figure 7: Negotiation with an easy opponent (Hard-Headed) in *Amsterdam Party* domain based on hard source task.

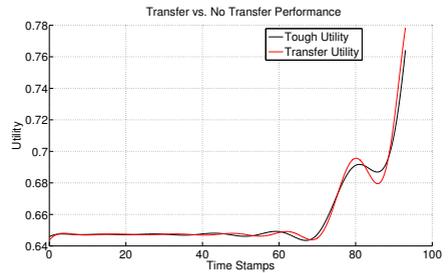


Figure 8: Negotiation with an easy opponent (Hard-Headed) in *Travel* domain based on hard source transferred to the target task.

**TNT** is the first transfer learning agent in negotiation settings. It tries to use already learned behavior in a source task in order to bias and improve learning against a target agent. The first important point is that **TNT** is a function approximation independent platform. In this work we chose to focus on GPs as we believe that the sought function might be complex and hard to picture. For that reason, we have employed a nonparametric functional space prior that is capable of capturing such properties as well as avoiding overfitting automatically. Using any other function approximation scheme such as neural networks, among others, is equally applicable. This increases the scope of applicability of **TNT** to any opponent modeling negotiation setting.

Transfer Learning has been deployed in a lot of interesting fields in machine learning. In the field of negotiation such a fame is still far-sought. Our presented results clearly demonstrate the applicability, and efficiency of transfer learning for negotiation tasks. To better understand the eight experimental results, we speculate that a weak preference to strategy relation, played a role in the success of the transfer scheme. Typically, in negotiation literature the relation between the agent’s preferences and the attained strategy is considered to be unimportant or even doesn’t exist. In other words, the played strategy is considered to be independent of the agent’s preferences. We disagree with this view and argue that such a relation is essential. To better clarify, no agent will produce an offer that is not to some extend influenced by its preferences as an upper bound in a successful negotiation session. We do acknowledge, however, that this relation is weak and that the preference-strategy is not a strong coupling. Having the goal of learning the opponents preferences in the future, and given the positive transfer results, we speculate that this weak preference-strategy relation was one of the main reasons for the success of **TNT**. To better clarify, in our experiments the source and target tasks had similar preferences on different issues but used different strategies (e.g., tough, weak, et cetera.). Since the source and the target opponent agents share this common similarity, and due to the existence of the weak preference-strategy correlation, we expect that these different strategies are weakly influenced by the similar preferences and thus have a “common

ground". From that point view, the positive results of the transfer algorithm are not surprising but are rather expected.

## 6 Conclusions and Future Work

In this paper we have presented **TNT**, the first transfer learning framework in negotiation settings. **TNT** makes use of gaussian processes and a probabilistic strategy mixture in order to improve learning against a target task opponents. A set of eight experiments in two different negotiation domains were conducted to proof the applicability of the proposed framework. We speculate that the weak preference to strategy relationship is one of the main reasons for the success of **TNT**. It is worth noting that **TNT** is not restrictive to what function approximation technique to be used.

In our future work we plan to target the following two issues. First, we will conduct a deeper analysis of the preference to strategy relation to better understand the transfer behavior among different opponents. Second, we plan to extend **TNT** to concurrent as well as to multi-issue dependent negotiation settings.

## References

- [1] Haitham Bou Ammar, Karl Tuyls, Matthew E. Taylor, Kurt Driessens, and Gerhard Weiss. Reinforcement learning transfer via space coding. In *Proceedings of 11th International Conference on Adaptive Agents and Multi-agent Systems (AAMAS 2012)*, Valencia, Spain, June 2012.
- [2] Jakub Brzostowski and Ryszard Kowalczyk. Predicting partner's behaviour in agent negotiation. In *Proceedings of the Fifth Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 355–361, New York, NY, USA, 2006. ACM.
- [3] Réal Carbonneau, Gregory E. Kersten, and Rustam Vahidov. Predicting opponent's moves in electronic negotiations using neural networks. *Expert Syst. Appl.*, 34:1266–1273, February 2008.
- [4] Siqi Chen and Gerhard Weiss. An efficient and adaptive approach to negotiation in complex environments. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI2012)*, Montpellier, France, 2012. IOS Press.
- [5] Robert M. Coehoorn and Nicholas R. Jennings. Learning on opponent's preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of the 6th Int. conf. on Electronic commerce*, ICEC '04, pages 59–68, New York, NY, USA, 2004. ACM.
- [6] K. Hindriks, C. Jonker, S. Kraus, R. Lin, and D. Tykhonov. Genius: negotiation environment for heterogeneous agents. In *Proceedings of AAMAS 2009*, pages 1397–1398, 2009.
- [7] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning.
- [8] Howard Raiffa. *The art and science of negotiation*. Harvard University Press Cambridge, Mass, 1982.
- [9] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [10] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982.
- [11] Sabyasachi Saha, Anish Biswas, and Sandip Sen. Modeling opponent decision in repeated one-shot negotiations. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, AAMAS '05, pages 397–403, New York, NY, USA, 2005. ACM.

# Algorithms for Basic Compliance Problems

Silvano Colombo Tosatto<sup>a</sup>  
Pierre Kelsen<sup>a</sup>

Marwane El Kharbili<sup>a</sup>  
Qin Ma<sup>a</sup>

Guido Governatori<sup>b</sup>  
Leendert van der Torre<sup>a</sup>

<sup>a</sup> *University of Luxembourg, Luxembourg*

<sup>b</sup> *NICTA, Australia*

## Abstract

The present paper focus on the problems to verify compliance for global achievement and maintenance obligations. We first introduce the elements needed to identify and study this fragment of compliance, such as processes and obligations. Afterwards we define the procedures and the algorithms to efficiently deal with this fragment of the compliance problem. Both algorithms proposed in the paper belongs to the **P** complexity class.

## 1 Introduction

Compliance initiatives are becoming more and more important in enterprises with the increase of the number of regulatory frameworks explicitly requiring businesses to show compliance with them. Most compliance solutions are ad hoc solutions and typically are time consuming to implement and to maintain. A classification of compliance in both preventive and detective activities is proposed in [10]. Auditing is a typical example of a detective activity. Preventive solutions, on the other hand, consider the activities to be done to achieve business objectives and their interactions with and the impact on them of the obligations and prohibitions imposed on a business by a normative framework. The proposal in [7] advances a compliance-by-design methodology. The methodology is based on the use of business process models to describe the activities of an enterprise and to couple them with formal specifications of the regulatory frameworks regulating the business. Business process models describe the activities to be done, and the order in which the task can be executed. Several approaches to handle compliance and to formalize normative requirements, based on different logical formalisms have been proposed ( see for example [5, 9, 3]).

The aim of this paper is not to propose yet another formalism for business process compliance, but to offer two efficient algorithms to deal with its most basic problems. In doing so, it allows scholars to reuse these basic algorithms in more complex frameworks thanks to their low computational complexity. In [2] it is shown that, in general, even for ‘well behaved’ classes of processes (i.e., structured processes), checking whether a process is compliant with a (formalised) legislation, is computationally hard. In the rest of paper we show how to use the abstract framework, to identify classes of compliance problems for which efficient solutions are possible.

The paper is structured as follows: Section 2 defines the compliance problem by introducing the definitions of processes and obligations. Section 3 describes the algorithms and their complexity. Section 4 concludes the paper.

## 2 Background

The scope of this paper is to decide wether a process is compliant with a given global obligation. In this section we formally describe processes and global obligations.

### 2.1 Processes

A process models a collection of methods to perform an activity. For instance an activity can be preparing coffee, a process modeling such activity would comprehend many ways to prepare coffee, such as using

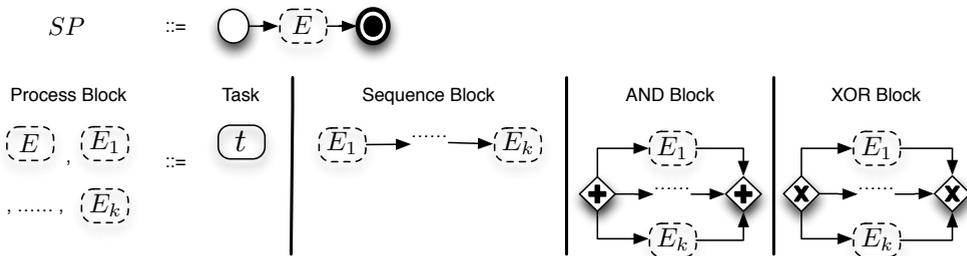
lyophilized coffee, brewing it, etc.

A process is composed of tasks and coordinators. Tasks represents the actions that can be done during the execution of a process. For instance considering the process that models how to prepare coffee, an action can be to heat the water. Coordinators are used to define the valid executions of a process. For instance coordinators can define that a certain task has to be done before another one or which tasks are mutually exclusive. Arrows connecting the elements of a process identify a general order in which the elements can be executed.

To represent a process we use a fragment of BPMN<sup>1</sup>. The fragment considered uses only AND and XOR coordinators in addition to start and end. The AND coordinator is used to coordinate tasks which can be concurrently executed. The XOR is used to define which tasks are mutually exclusive. AND and XOR coordinators consist in blocks of tasks within the process which are enclosed between two coordinators of the same type.

We consider only processes which do not contain cycles and are structured. A process is structured if it consists of hierarchically nested blocks as depicted by the following classification.

**Definition 1 (Process)** A structured business process  $P$  is a business process generated by the following grammar given in the format of an graphical extension of BNF (with the vertical lines indicating alternative for the right hand side):



The coordinator  $\circ$  is called start and the coordinator  $\bullet$  is called end. The coordinator  $\oplus$  is called ANDsplit in case of multiple outgoing arrows and ANDjoin in case of multiple incoming arrows. A pair of ANDsplit and ANDjoin coordinators groups a set of sub-blocks indicating a logical relationship to activate all the sub-blocks concurrently. Finally, the coordinator  $\otimes$  is called XORsplit in case of multiple outgoing arrows and XORjoin in case of multiple incoming arrows. A pair of XORsplit and XORjoin coordinators groups a set of sub-blocks indicating a logical relationship to activate exactly one of the sub-blocks, chosen arbitrarily.

We assume that all the tasks in a structured business process carry a distinct identity that constitutes a key part of the label of a task. Therefore, a task  $t$  can directly be referenced by its label  $t$ . Similarly, (process) block identities are also distinct hence a block  $\{E\}$  can directly be referenced by its label  $E$ . As a consequence, for simplicity, we also allow a textual way to reference the graphical representation of structured business processes.

**Example 1** In Fig. 1 we provide an example of a process containing four tasks labeled  $t_1, \dots, t_4$ . Within the process it is shown an XOR block containing in different branches the tasks  $t_1$  and  $t_2$ . The XOR block is nested within an AND block, forming one of its branches and task  $t_3$  forming the other one. The AND block is preceded by the start coordinator and followed by task  $t_4$  which in turn is followed by the end coordinator.

Given a process modeling an activity, an execution of such a process represents one way to perform it. An execution is a valid serialization of a subset of tasks composing the process. A serialization is considered valid if it starts from the start coordinator and terminates at the end. In addition a valid serialization has to comply with the semantics of the coordinators and the connections between the tasks.

A process is defined as  $P = \text{start } E \text{ end}$ . An execution of  $P$  is equivalent to executing the block  $E$  within start and end. Thus we will provide the formal semantics for executing blocks which can be used for process execution as well.

<sup>1</sup>Business Process Model Notation, Version 2.0, <http://www.omg.org/spec/BPMN/2.0>

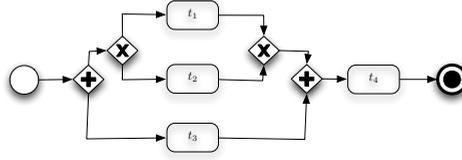


Figure 1: Example of a process

**Definition 2 (Block Execution)** A process block  $E$  can be serialized into a set of finite sequences of tasks, written  $\Sigma(E)$ , defined by the following structural recursion. We call each sequence in  $\Sigma(E)$  an execution of  $E$ , ranged over by  $\epsilon$ .

1.  $E = t$ :  $\Sigma(E) = \{(t)\}$ ;
2.  $E = \text{SEQ}(E_1, \dots, E_k)$ :  $\Sigma(E) = \{\epsilon_1; \dots; \epsilon_k \mid \epsilon_1 \in \Sigma(E_1), \dots, \epsilon_k \in \Sigma(E_k)\}$ , where  $;$  stands for sequence concatenation.
3.  $E = \text{XOR}(E_1, \dots, E_k)$ :  $\Sigma(E) = \Sigma(E_1) \cup \dots \cup \Sigma(E_k)$ ;
4.  $E = \text{AND}(E_1, \dots, E_k)$ :  $\Sigma(E) = \{(t_1, \dots, t_n)\}$  such that
  - (a)  $\forall i, 1 \leq i \leq k, \exists \epsilon_i \in \Sigma(E_i)$  such that  $\{t_1, \dots, t_n\} = \bigcup_{1 \leq i \leq k} \text{Tasks}(\epsilon_i)$
  - (b)  $\forall E_i \in E = \text{AND}(E_1, \dots, E_k), t_h, t_j \in E_i \mid t_h < t_j \rightarrow \forall \epsilon \in \Sigma(E), t_h > t_j$ .

Namely  $\Sigma(E)$  is the set of sequences each of which merges a sequence of  $\Sigma(E_1), \dots$ , and of  $\Sigma(E_k)$ . Merging a set of sequences gives rise to a sequence that includes all the elements of the operand sequences. Moreover, the ordering in the result sequence should be compatible with the ordering in the operand sequences.

In an arbitrary process and its possible executions. If the process is conform with Definition 1, then a task belonging to the process appears in at least one of its executions. This means that each task contained in a process has the possibility to be executed as stated in the following lemma.

**Lemma 1 (Block Execution)** Given a process block  $E$  and a task  $t$  in  $E$ ,  $\exists \epsilon \in \Sigma(E)$  such that  $t \in \epsilon$ .

**Example 2** Taking into account the process in Fig. 1 as  $P = \text{start } E \text{ end}$ . We have that  $\Sigma(E) = \{\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4\}$  where  $\epsilon_1 = (t_1, t_3, t_4)$ ,  $\epsilon_2 = (t_2, t_3, t_4)$ ,  $\epsilon_3 = (t_3, t_1, t_4)$  and  $\epsilon_4 = (t_3, t_2, t_4)$ .  $\Sigma(E)$  contains the four possible executions of the process  $P$ . An execution not contained in  $\Sigma(E)$ , like  $\epsilon_5 = (t_3, t_4, t_1)$ , is not a valid execution of  $P$ . In this particular case one of the reasons why  $\epsilon_5$  is not a valid execution is because after  $t_4$  the task  $t_1$  is executed, which is not possible because  $t_1$  belongs to an XOR block nested in an AND block that precedes the task  $t_4$  in a sequence block.

The state of the process changes while executing the tasks. We represent the state of a process as an incomplete consistent set of literals. Given a language, a set is called incomplete if there exists a literal of the language such that neither literal nor its complement belongs to the set.

**Definition 3 (Consistent Literal Set)** Given a literal  $l$ , let  $\bar{l}$  be its complement. A set of literals  $L$  is consistent if and only if it does not contain  $l$  and  $\bar{l}$  at the same time for every literal  $l \in L$ .

**Example 3** In a language of literals containing  $\{\alpha, \beta, \gamma\}$ , the following states:  $L_1 = \{\alpha, \bar{\beta}\}$ ,  $L_2 = \{\bar{\alpha}, \bar{\beta}, \gamma\}$ ,  $L_3 = \{\alpha, \bar{\alpha}, \beta\}$ .  $L_1$  is an incomplete state because it does not contain either  $\gamma$  or its complement.  $L_1$  is also consistent because it does not contain a literal and its complement.  $L_2$  is a complete state because it contains all the literals or their complement belonging to the alphabet and  $L_3$  is inconsistent because it contains both  $\alpha$  and  $\bar{\alpha}$ .

Executing a task can change the current state of the process. Such changes depend on a consistent set of literals associated to the task being executed. We refer to a task with an associated set of literals as *annotated task*. The set of literals of an annotated task indicates the postconditions that have to hold after the task is executed. A process containing annotated tasks is called an *annotated process*.

**Definition 4 (Annotated Process)** An annotated process is a pair:  $(P, \text{ann})$ , where  $P$  is a process and  $\text{ann} : T \rightarrow 2^{\mathcal{L}}$  is a function from the set  $T$  of  $P$  to consistent sets of literals of a language  $\mathcal{L}$ .

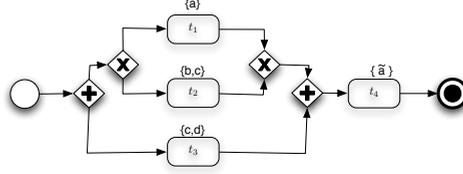


Figure 2: Example of an annotated process

**Example 4** Fig. 2 resumes the previous example shown in Fig. 1 by including annotations for its tasks. We can see that after executing task  $t_1$ , the literal  $a$  has to hold in the successive state. Annotations are not limited to single literals: tasks  $t_2$  and  $t_3$  are both annotated by multiple literals.

The execution state of a process has to be kept consistent. This means that after the execution of an annotated task, the literals in the set associated to such task must hold in the state but the state has to be kept consistent. To allow such behavior, before updating the current state we remove from it the literals which could cause inconsistencies with the ones introduced by the task execution. After this step the state can be updated by including the literals in the set of the annotated task. Being the set of literals introduced consistent by definition, the result is still a consistent set.

**Definition 5 (Literal set update)** Given two consistent sets of literals  $L_1$  and  $L_2$ , the update of  $L_1$  with  $L_2$  is a set of literals defined as follows:

$$L_1 \oplus L_2 = L_1 \setminus \{\tilde{l} \mid l \in L_2\} \cup L_2$$

**Example 5** This example shows how the state of a process is updated after executing a task. Given three sets of literals:  $L_1 = \{a, b\}$ ,  $L_2 = \{a, b, c\}$  and  $L_3 = \{\tilde{c}\}$ . In case of  $L_1 \oplus L_3$  the result is the set  $\{a, b, \tilde{c}\}$  which represent the union of  $L_1$  and  $L_3$ . Differently if we consider  $L_2 \oplus L_3$  the result is again  $\{a, b, \tilde{c}\}$  but this time the result is not equivalent to  $L_2 \cup L_3$  because  $L_3$  contains  $\tilde{c}$  that is the complement of one of the literals in  $L_2$ . The literal  $c$  is discarded from  $L_2$  before joining it with  $L_3$  so that the result is a consistent set. We can notice that  $\oplus$  is not commutative because in the case  $L_3 \oplus L_2$  the result would be  $\{a, b, c\}$ .

During one of its possible executions, a process typically goes through several states. Each of these states can be associated to the execution of one of the annotated tasks belonging to the execution. We call a *trace* such sequence of states and tasks.

**Definition 6 (Trace)** The trace  $\theta$  corresponding to an execution  $\epsilon = (t_1, \dots, t_k)$  of an annotated process  $(P, \text{ann})$  is a finite sequence of pairs of the form  $((t_1, L_1), \dots, (t_k, L_k))$ , where  $L_1, \dots, L_k$  are sets of literals such that:

1.  $L_1 = \text{ann}(t_1)$ ;
2.  $L_{i+1} = L_i \oplus \text{ann}(t_{i+1})$ , for  $1 \leq i < k$ .

We write  $\Theta((P, \text{ann}))$  to denote the set of traces of an annotated process, and let  $\theta$  range over it.

**Lemma 2 (Trace)** Traces and execution of a process are in one to one correspondence.

**Example 6** This example shows the traces of the annotated process  $(P, \text{ann})$  illustrated in Fig. 2. In the following table we show for each execution of  $P$  the corresponding trace. Each trace is represented as a sequence of pairs where every pair represents the task executed and the state holding after its execution.

execution	trace
$(t_1, t_3, t_4)$	$((t_1, \{a\}), (t_3, \{a, c, d\}), (t_4, \{\tilde{a}, c, d\}))$
$(t_2, t_3, t_4)$	$((t_2, \{b, c\}), (t_3, \{b, c, d\}), (t_4, \{\tilde{a}, b, c, d\}))$
$(t_3, t_1, t_4)$	$((t_3, \{c, d\}), (t_1, \{a, c, d\}), (t_4, \{\tilde{a}, c, d\}))$
$(t_3, t_2, t_4)$	$((t_3, \{c, d\}), (t_2, \{b, c, d\}), (t_4, \{\tilde{a}, b, c, d\}))$

## 2.2 Obligations

A trace is said to be compliant if it respects a given *global obligation*. We use a subset of Process Compliance Logic (PCL) [6] to specify the global obligations.

A global obligation is an obligation that holds from the start till the end of a process. There are two different types of global obligations: achievement and maintenance.

**Definition 7 (Global Obligations)** *Given a literal  $l$  as the condition of a global obligation  $\mathcal{O}$ , we represent the two types of obligations as follows:*

$$\begin{array}{l} \mathcal{O} ::= O^a(l) \text{ Achievement Obligation} \\ \quad | O^m(l) \text{ Maintenance Obligation} \end{array}$$

The condition of an achievement obligation, has to be verified in at least one state of the trace between the start and the end of a trace. An achievement obligation is violated if no state before the end of the trace satisfies the condition.

For maintenance obligations, every state between the start and the end of a trace has to fulfill the condition. A maintenance obligation is violated as soon as a state does not verify the condition.

**Definition 8 (Global Obligation Fulfillment)** *Given a global obligation  $\mathcal{O}$  and a trace  $\theta = ((t_1, L_1), \dots, (t_k, L_k))$ ,  $\theta$  fulfills  $\mathcal{O}$  ( $\theta \vdash \mathcal{O}$ ) iff:*

- $\mathcal{O} = O^a(l) \theta \vdash O^a(l)$  iff  $\exists L_i, 1 \leq i \leq k | l \in L_i$ .
- $\mathcal{O} = O^m(l) \theta \vdash O^m(l)$  iff  $\forall L_i, 1 \leq i \leq k | l \in L_i$ .

## 2.3 Process Compliance

Checking if a process is compliant with an obligation can return three different results. A process is fully compliant if every trace of the process is compliant with the obligation. A process is partially compliant if there exists a trace compliant with the obligation. If none of the traces of a process are compliant with the obligation, then the process is not compliant.

**Definition 9 (Process Compliance)** *Given an annotated process  $(P, \text{ann})$  and a global obligation  $\mathcal{O}$*

- **Full compliance**  $(P, \text{ann}) \models^f \mathcal{O}$  iff  $\forall \theta \in \Theta((P, \text{ann})), \theta \vdash \mathcal{O}$ .
- **Partial compliance**  $(P, \text{ann}) \models^p \mathcal{O}$  iff  $\exists \theta \in \Theta((P, \text{ann})), \theta \vdash \mathcal{O}$ .
- **Not compliant**  $(P, \text{ann}) \not\models \mathcal{O}$  iff  $\exists \theta \in \Theta((P, \text{ann})), \theta \vdash \mathcal{O}$ .

## 3 Algorithms and Complexity

In this section we present the algorithms to verify the compliance of a process with respect to a global obligation. We design two algorithms, one for each type of global obligation.

### 3.1 Algorithm for Global Achievement Obligations

The algorithm for achievement obligations uses the function *Task Removal*. This function is used to remove a set of tasks from a process. By removing some tasks, the executions that contain that task are no longer allowed. In some cases by removing one or more tasks from a block it is possible that no executions remain available, if this is the case the function does not return a process block but  $\perp$ .

**Definition 10 (Task Removal)** *Given a process  $P = \text{start } E \text{ end}$  and a set of tasks  $T$ , task removal  $R(E, T)$  returns either a new process block  $E'$  or  $\perp$  as follows:*

1.  $E = t$ : if  $t \in T$  then return  $\perp$  else return  $E$ ;
2.  $E = \text{SEQ}(E_1, \dots, E_k)$ :  
 if  $\exists i, 1 \leq i \leq k$  such that  $R(E_i, T) = \perp$  then return  $\perp$ ,  
 else return  $\text{SEQ}(R(E_1, T), \dots, R(E_k, T))$ ;
3.  $E = \text{XOR}(E_1, \dots, E_k)$ :  
 if  $\forall i, 1 \leq i \leq k, R(E_i, T) = \perp$  then return  $\perp$ ,  
 else if  $\exists! i, 1 \leq i \leq k$  such that  $R(E_i, T) \neq \perp$  then return  $R(E_i, T)$ ,  
 else return  $\text{XOR}(R(E_{m_1}, T), \dots, R(E_{m_n}, T))$  for all  $\forall E_{m_j} \in \{E_1, \dots, E_k\} | R(E_{m_j}, T) \neq \perp$ ;

4.  $E = \text{AND}(E_1, \dots, E_k)$ :  
**if**  $\exists i, 1 \leq i \leq k$  such that  $R(E_i, T) = \perp$  **then return**  $\perp$ ,  
**else return**  $\text{AND}(R(E_1, T), \dots, R(E_k, T))$ .

**Lemma 3 (Task Removal)** *Given a process block  $E$  and a set of tasks  $T$  in this block,*

1.  $R(E, T) = \perp$  *iff*  $\forall \epsilon_i \in \Sigma(E), \epsilon_i \cap T \neq \emptyset$ ;
2. *otherwise*,  $R(E, T) = E'$  where:
  - (a)  $\Sigma(E') \subseteq \Sigma(E)$ ;
  - (b)  $\forall \epsilon_i \in \Sigma(E), \epsilon_i \cap T = \emptyset$  *iff*  $\epsilon_i \in \Sigma(E')$ , and  $\forall \epsilon_i \in \Sigma(E'), \epsilon_i \cap T = \emptyset$ .

In other words:  $E'$  contains exactly the traces of  $E$  that do not have tasks in  $T$ .

**Algorithm 1** *Given an annotated process  $(P, \text{ann})$  and a global achievement obligation  $O^a(l)$ , this algorithm returns whether  $(P, \text{ann})$  is compliant with  $O^a(l)$ .*

- 1: *Suppose*  $P = \text{start } E \text{ end.}$ ;
- 2: **if**  $\forall t$  in  $E, l \notin \text{ann}(t)$  **then**
- 3:   **return**  $(P, \text{ann}) \not\models O^a(l)$ ;
- 4: **else**
- 5:   **if**  $R(E, \{t \mid t \text{ is a task in } E \text{ and } l \in \text{ann}(t)\}) = \perp$  **then**
- 6:     **return**  $(P, \text{ann}) \not\models O^a(l)$ ;
- 7:   **else**
- 8:     **return**  $(P, \text{ann}) \models O^a(l)$ ;
- 9:   **end if**
- 10: **end if**

Due to the nature of an achievement obligation  $O^a(l)$ , it is satisfied when a task whose annotation contains the condition  $l$  of the obligation is executed. By removing all the tasks containing  $l$  in their annotations, the remaining executions are the ones which do not fulfill the obligation. If there are no possible executions remaining, this means that every execution has to go through at least a task having  $l$  annotated, thus the process is fully compliant with the obligation. In case there are no tasks having  $l$  in their annotation, then no execution can fulfill the obligation and the process is not compliant. At last if some tasks are removed and some possible executions remain, then the process is partially compliant.

**Complexity of Algorithm 1:** Assuming that the size of each annotation is  $\mathcal{O}(1)$ , i.e. independent of the number of tasks. The time of Algorithm 1 is dominated by the time for the task removal algorithm which is linear in the number of tasks of the process.

### 3.2 Algorithm for Global Maintenance Obligations

We first introduce the notion of *first tasks*, which are the set of tasks of a process that can be scheduled at the beginning of an execution.

**Definition 11 (First Task(s))** *Given a process block  $E$   $\text{First}(E)$ , returns a set of tasks as follows:*

- $E = t: \{t\}$ ;
- $E = \text{SEQ}(E_1, \dots, E_k)$  where  $k \geq 2$ :  $\text{First}(E_1)$ ;
- $E = \text{AND}(E_1, \dots, E_k)$  where  $k \geq 2$ :  $\bigcup_{i=1}^k \text{First}(E_i)$ ;
- $E = \text{XOR}(E_1, \dots, E_k)$  where  $k \geq 2$ :  $\bigcup_{i=1}^k \text{First}(E_i)$ .

Given a block  $E$  and a task  $t \in \text{First}(E)$ , let  $X$  denote the set of executions in  $E$  that have  $t$  as the first task. The function *Task Rooting* returns a subset of the executions contained in  $X$ . In Lemma 4 we provide a sketch of a proof showing that the approximation considered by *Task Rooting* does not affect the result of checking compliance for maintenance obligations.

**Definition 12 (Task Rooting)** *Given a process block  $E$  and a task  $t \in \text{First}(E)$ , task rooting  $F(E, t)$  returns a new process block as follows:*

1.  $E = t$ : *return*  $E$ ;
2.  $E = \text{SEQ}(E_1, \dots, E_k)$ : *return*  $\text{SEQ}(F(E_1, t), E_2, \dots, E_k)$ ;
3.  $E = \text{XOR}(E_1, \dots, E_k)$ : *return*  $F(E_p, t)$  where  $E_p \in \{E_1, \dots, E_k\}$  and  $t \in E_p$ .

4.  $E = \text{AND}(E_1, \dots, E_k)$ : return  $\text{SEQ}(F(E_p, t), \text{AND}(E_{i_1}, \dots, E_{i_{k-1}}))$ , where  $\{i_1, \dots, i_{k-1}, p\} = \{1, \dots, k\}$  and  $t \in E_p$ .

**Lemma 4 (Task Rooting)** *Let  $E$  be a block,  $t$  be a task such that  $t \in \text{First}(E)$  and,  $X$  be the set of executions of  $E$  that start with  $t$ . We denote with  $\theta \in \Theta_X$  a trace associated to an  $\epsilon \in X$  according to the annotation an ann:*

$$\begin{aligned} ((\text{start}, (F(E, t), \text{ann}), \text{end})) \vdash^E O^m(l)) &\Leftrightarrow (\forall \theta \in \Theta_X, \theta \vdash O^m(l)) \\ ((\text{start}, (F(E, t), \text{ann}), \text{end})) \vdash^L O^m(l)) &\Leftrightarrow (\exists \theta \in \Theta_X, \theta \vdash O^m(l)) \\ ((\text{start}, (F(E, t), \text{ann}), \text{end})) \not\vdash O^m(l)) &\Leftrightarrow (\forall \theta \in \Theta_X, \theta \not\vdash O^m(l)) \end{aligned}$$

**Proof 1** *Given a process  $P$ , we know that  $X$  contains all executions of  $P$  starting with an arbitrary task  $t$ . The function task rooting returns an approximation of the set  $X$  only in the case where  $t$  belongs to an AND block. The executions that are contained in the process block returned by task rooting are the one which have as a prefix the brach of the AND block starting with  $t$ . The executions lost by task rooting are the ones where some tasks from other branches in the AND block are interleaved with the ones belonging to the branch containing  $t$ . We can focus on the serialization of the AND block because it is where some executions are lost.*

*We distinguish now two cases:  $l \notin \text{ann}(t)$  and  $l \in \text{ann}(t)$ . In the first case both the executions in  $X$  and the ones given by task rooting are not compliant according to Definition 8. In the second case we have to analyze the remainder tasks in the AND block.*

*In case none of the remainder tasks annotates  $\tilde{l}$ , then we can safely that in both cases the AND block is fully compliant with the maintenance obligation. In case where some of the tasks contain in their annotation  $\tilde{l}$ , we have to analyze two cases: the first where such tasks are not avoidable, i.e. these tasks are not within an XOR block, which means that  $l$  would stop holding due to the execution of one of these tasks, thus the AND block would be not compliant in this case both in  $X$  and task rooting, because at least one task containing  $\tilde{l}$  had to be executed. In case such tasks are avoidable, thus both for  $X$  and task rooting exists at least an execution which is compliant with the maintenance obligation, making the AND block partially compliant with the obligation.*

*It is not necessary to analyze the part of the executions following the AND block because no more approximations are involved, thus after having shown that the approximation on the AND block does not alter the result, we can say that the result obtainable by checking a maintenance obligation after applying task rooting, would be the same as checking it on the set  $X$ .*

**Algorithm 2** *Given an annotated process  $(P, \text{ann})$  and an atomic maintenance obligation  $O^m(l)$ , this algorithm returns whether  $(P, \text{ann})$  is compliant with  $O^m(l)$ .*

```

1: Suppose  $P = \text{start } E \text{ end}$ ;
2:  $\text{res} = (P, \text{ann}) \not\vdash O^m(l)$ ;
3:  $T_F = \text{First}(E)$ ;
4:  $T_{\tilde{l}} = \{t \in P : \tilde{l} \in \text{ann}(t)\}$ ;
5: if  $\forall t \text{ in } T_F, l \in \text{ann}(t)$  and  $T_{\tilde{l}} == \emptyset$  then
6:   return  $(P, \text{ann}) \vdash^E O^m(l)$ ;
7: else
8:   for each  $t \in T_F$  such that  $l \in \text{ann}(t)$  do
9:     if  $R(F(E, t), T_{\tilde{l}}) \neq \perp$  then
10:      return  $(P, \text{ann}) \vdash^E O^a(l)$ ;
11:   end if
12: end for each
13: return  $(P, \text{ann}) \not\vdash O^m(l)$ ;
14: end if
    
```

Algorithm 2 identifies the set of the tasks that can appear as first in the possible executions of the process in analysis. From such set the algorithm identifies which executions has the possibility to be compliant by starting with a task having  $l$  annotated. For each execution that can be compliant, *Task Removal* is used to verify that they don't contain a task with  $\tilde{l}$  annotated.

**Complexity of Algorithm 2:** Both  $R$  and  $F$  can be computed in time  $O(n)$  where  $n$  is the number of tasks in  $E$ . Thus the call to  $R$  on line 9 can also be computed in time  $O(n)$ . Assuming each annotation has size  $O(1)$  we then see that the overall complexity is  $O(n^2)$ .

## 4 Conclusion

Business process compliance received increased attention in the field of business process modeling in the past few years. The majority of approaches propose some logics for compliance (e.g., deontic logic [5], linear temporal logic [11], clause based logic/logic programming [3, 4], extensions of BPMN languages [1]). However, to the best of our knowledge, this is a first systematic investigation on the complexity of business process compliance. [6] provides a linear time algorithm to check whether a single trace is compliant, and [8] gives approximate solutions in linear time. [2] shows that the problem of checking whether a process is compliant or not is computationally infeasible. In this paper we have identified some tractable sub-problems and their solutions.

As future work we plan to identify further problems and provide solutions by integrating our algorithms in an *abstract framework* or designing new ones if needed.

## References

- [1] Ahmed Awad, Gero Decker, and Mathias Weske. Efficient compliance checking using bpmn-q and temporal logic. In Marlon Dumas, Manfred Reichert, and Ming-Chien Shan, editors, *BPM*, volume 5240 of *Lecture Notes in Computer Science*, pages 326–341. Springer, 2008.
- [2] Silvano Colombo Tosatto, Guido Governatori, Pierre Kelsen, and Leendert van der Torre. Business process compliance is hard. Technical report, NICTA, 2012.
- [3] Aditya Ghose and George Koliadis. Auditing business process compliance. In Bernd J. Krämer, Kwei-Jay Lin, and Priya Narasimhan, editors, *ICSOC*, volume 4749 of *Lecture Notes in Computer Science*, pages 169–180. Springer, 2007.
- [4] Guido Governatori, Jörg Hoffmann, Shazia Wasim Sadiq, and Ingo Weber. Detecting regulatory compliance for business process models through semantic annotations. In Danilo Ardagna, Massimo Mecella, and Jian Yang, editors, *Business Process Management Workshops*, volume 17 of *Lecture Notes in Business Information Processing*, pages 5–17. Springer, 2008.
- [5] Guido Governatori, Zoran Milosevic, and Shazia Sadiq. Compliance checking between business processes and business contracts. In Patrick C. K. Hung, editor, *10th International Enterprise Distributed Object Computing Conference (EDOC 2006)*, pages 221–232. IEEE, 2006.
- [6] Guido Governatori and Antonino Rotolo. Norm compliance in business process modeling. In *Proceedings of the 4th International Web Rule Symposium: Research Based and Industry Focused (RuleML 2010)*, volume 6403 of *LNCS*, pages 194–209. Springer, 2010.
- [7] Guido Governatori and Shazia Sadiq. The journey to business process compliance. In Jorge Cardoso and Wil van der Aalst, editors, *Handbook of Research on BPM*, chapter 20, pages 426–454. IGI Global, 2009.
- [8] Jörg Hoffmann, Ingo Weber, and Guido Governatori. On compliance checking for clausal constraints in annotated process models. *Information Systems Frontiers*, 14(2):155–177, 2012.
- [9] Dumitru Roman and Michael Kifer. Reasoning about the behaviour of semantic web services with concurrent transaction logic. In *VLDB*, pages 627–638, 2007.
- [10] Shazia Sadiq and Guido Governatori. Managing regulatory compliance in business processes. In Jan van Brocke and Michael Rosemann, editors, *Handbook of Business Process Management*, volume 2, chapter 8, pages 157–173. Springer, Berlin, 2010.
- [11] Wil M. P. van der Aalst, Maja Pesic, and Helen Schonenberg. Declarative workflows: Balancing between flexibility and support. *Computer Science - R&D*, 23(2):99–113, 2009.

# An Interactive Learning Approach to Histology Image Segmentation

Michaël Derde<sup>a</sup>   Laura Antanas<sup>a</sup>   Luc De Raedt<sup>a</sup>   Fabian Guiza Grandas<sup>b</sup>

<sup>a</sup> *Katholieke Universiteit Leuven, Department of Computer Science*

<sup>b</sup> *Katholieke Universiteit Leuven, Laboratory of Intensive Care Medicine*

## Abstract

Histology image analysis using computer-aided diagnosis systems has become increasingly important during the last years. One reason is the need to alleviate the heavy workload of medical experts. In this paper, we introduce a general purpose framework which is able to solve histology analysis problems that are not restricted to a specific type of tissue or task, exploit local information in microscopical images, interact with medical experts and iteratively consider direct user feedback. The framework is general enough to learn models that can adapt to several learning tasks and can detect several types of medical interest regions. We evaluate our framework on real-world datasets collected from patients in the intensive care unit. We considerably outperform image processing techniques commonly used in such medical imaging tasks.

## 1 Introduction

Histology is the anatomical study of the microscopic structure of tissues. It is regarded as a gold standard for clinical diagnosis of diseased tissue (e.g., cancer) and for the identification of therapy effects [10]. Histological analysis is performed by examining a thin section of tissue under a microscope [13, 20, 16], after applying a sequence of procedures for tissue preparation: fixation, dehydration, clearing, infiltration, embedding, sectioning and staining [14]. It reveals information about cells and tissue with a high level of detail. Despite the great care taken in their preparation, histology images are prone to several artifacts, e.g., folding of the tissue section, overlap among cell boundaries, noise introduced by the microscope or slides, blurry sections, etc. As a result, analysis of histology tissues remains most of the time a manual endeavor which relies heavily on the expertise of the medical expert. This manual work is however very time consuming and prone to subjective interpretation. Therefore, computer-assisted diagnosis (CAD) systems are becoming crucial in histology analysis, as they could automatically identify regions of medical interest. Their main advantage is the ability to provide immediate results in a consistent and objective manner, thereby reducing the workload of the medical experts.

With few exceptions [24], current image analysis tools focus on specific tasks, such as nuclei and cell counting, and lack the flexibility of dealing with a variety of tissue types that might be of interest to medical experts. Moreover, most systems make use of traditional image processing approaches such as global thresholding, region growing, region splitting and merging, and active contours (for more details see [18, 28]). Their main drawback is that they fail to account for local variations (e.g., brightness, staining intensity) within a single image that are introduced by the microscope or lightening conditions.

In this paper we present a new CAD framework which learns to automatically detect regions of interest in histology images. It overcomes drawbacks of the current approaches by combining an interactive learning technique that adapts to the specific user-defined medical task, with a local approach that takes into account local variations of particular regions of interest. Instead of the basic supervised learning paradigm in which the expert is asked to label examples and then a predictor is learned from these targets, without other explicit interaction, our framework uses expert knowledge to interactively feed training instances to the learning system. Once a new instance has been added by the expert, a completely new model is built from scratch, as a new supervised learning step. In this way, by changing the learning targets after each iteration, our framework can incorporate real-time feedback for the current model predictions, reducing training time and data. Once the model has been trained, it can be used to automatically process any amount of images. We formalize the supervised learning problem as a regression task and we employ regression trees to represent

the learned model. The training instances used to build the model are slices sampled from full histology images. They are segmented by the medical expert via local thresholding. The user-defined thresholds become regression targets in the learning phase.

In the remaining sections we present related work in automated analysis of histology images and describe the methodology employed by our interactive framework. Next, we present an experimental evaluation of our framework on real-world data collected from patients in the intensive care unit and discuss the results. Finally, we provide concluding remarks and perspectives for future work.

## 2 Related work

This section discusses related work on automated analysis of histology images. A crucial task in histology analysis is to automatically segment the histology image in background regions and regions of interest (e.g., nuclei, cells, stained regions). A comprehensive survey of image segmentation techniques is presented in [18, 8]. A popular and simple segmentation approach is intensity or color thresholding, which can be further described as being either bi-level or multi-thresholding. Bi-level refers to the use of only one threshold value, where image pixels below (above) the value are considered as background (region of interest). The use of several thresholds allows multi-thresholding to segment an image into more than two types of regions [21]. Alternatives to thresholding techniques that were successfully applied to histological segmentation include perceptual grouping [27], region growing [19], fuzzy clustering [2], active contours [4] and energy-based methods [25]. These alternatives, however, are either too problem specific or too demanding computationally for a fast interactive framework.

Recently, considerable work on histological segmentation has focused on fully automated approaches that rely on machine learning techniques. They allow for large amounts of data to be processed. Based on descriptive features of the data interesting patterns are discovered. The extraction of meaningful features from histology images has been addressed in [1]. We employ such features in our framework, yet we also consider different ones. Some related machine learning approaches have been employed for histology images. One of them uses Markov Random Fields in a Bayesian formulation and has been employed to segment cancerous structures [29]. Another approach uses a bag of local Bayesian classifiers to classify pixels as belonging to cells or not [31] and thus, segment histology images. Finally, the work in [7] uses random forests to classify pixels as belonging to a fixed set of predefined classes. Different from these, our approach employs regression trees to learn meaningful thresholds which are then used to segment the image.

Other related techniques that incorporate the human expert into the learning loop are relevance feedback, preference elicitation and active learning. Relevance feedback improves current predictions by considering direct user feedback with respect to the relevance of past reactions [32]. Preference elicitation computes preferences from utility function estimation [6] and active learning asks the expert to analyze only instances that would be the most informative for the learning task [12]. Because the expert dynamically analyzes the instances, a single system can perform several learning tasks always using the same pre-computed set of features. While these techniques have been utilized in computer vision [23, 11], they remain largely unused in automated analysis of histology analysis. Differently, our framework uses expert knowledge to feed new training instances to the learning system, without specifying their degree of relevance and building, at each iteration, a completely new model from scratch.

Several software solutions are available for histology analysis. They include CellProfiler [5], CellTracer [30], CellTracker [22] and Imago [15]. These tools provide automatic segmentation of cells and nuclei and report statistics in similar terms about the image. They allow for batch processing, albeit they can not adapt parameter settings per individual image which remain static throughout the entire batch evaluation. Imago also provides an interactive segmentation method, yet it is restricted to bi-level thresholding. Thus, current available solutions suffer from two major drawbacks. First, they focus on a specific task, such as cell or nuclei segmentation, which hinders their use for other regions of interest. Second, they make use of global approaches both in analyses of one image and in batch processing, which means that varying lighting conditions or clinically relevant variations in staining intensity are largely ignored. Closely related is the work in [24], a learning and segmentation toolkit for image classification and segmentation which works iteratively in a similar way. However, differently, we propose a regression-based approach.

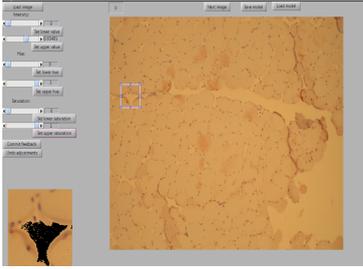


Figure 1: GUI interface. A slider is used to manually segment the regions of interest.

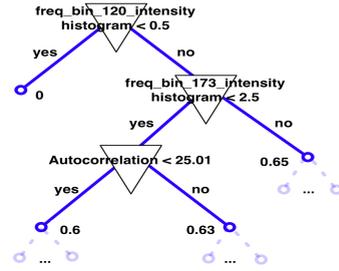


Figure 2: Regression tree model for nuclei detection in muscle tissue. The outcome is the local real-valued intensity threshold prediction.

### 3 Approach

In this section we describe our learning approach which overcomes the drawbacks in current approaches for detecting generic regions of interest in medical images. We start from randomly generated slices from the histology image. These slices are sections or crops of different sizes that are assumed independent from each other and that do not overlap. Our goal is to find the optimal segmentation of the areas of medical interest for each of these slices. We keep the well-known thresholding approach [17], however instead of manually selecting thresholds by considering the peaks of histograms for different image channels, we automatically learn a model to predict these thresholds.

There are several advantages of this approach. One is that the prediction relies totally on a set of features of the image slice which are the same independently of the problem considered. Another advantage is that the local threshold is objectively generated by the model according to the slice features and not on a subjectively selected threshold that is usually found between adjacent peaks of a histogram. Finally, there may be different image targets to threshold (e.g., staining intensity and hue) depending on the problem at hand. Employing a method which can automatically predict optimal thresholds for all targets considered makes our approach problem independent also with respect to the learning targets.

#### 3.1 Interactive data acquisition

One way commonly used in medical image processing to obtain regions of interest is to impose thresholds on image parameters such as the intensity, hue and saturation channels. By considering all channels, our approach is more precise and general for more complex problems. If solely intensity is used, making distinctions between two regions that have similar RGB values is difficult, if not impossible, while in the hue channel they may be more disjoint, allowing for a better segmentation. Therefore, we consider a *multi-target* setting, where the goal is to threshold not only the different image channels at the same time, but even more complex targets for each channel so as to obtain an even more precise segmentation.

Often, medical specialists need to identify more complex tissue components than nuclei or cells. These may include particular stained regions. We propose a general purpose *interactive* framework which interacts with the user and can adapt to specific problems by learning from user feedback. This means that the domain expert is able to choose which examples to label such that an optimal model for the specific task of interest is built. In particular, the medical expert interacts with the framework and presents to the algorithm predictions for each image slice. This interaction is performed via a slider which is properly adjusted by the user in a graphical interface (as shown in Figure 1). Once a new instance has been segmented by the expert, a completely new model that incorporates the new user feedback is built from scratch. The newly trained model segments the next user-selected instance, and if needed, the expert further refines the segmentation, by providing additional feedback. The final model can then be used to batch process the remaining images.

#### 3.2 Problem definition

We formalize the detection of areas of medical interest as a supervised learning problem: given a set of image slices  $X = \{\mathbf{x}_i\}_{i=1}^n$  labeled with classes  $Y = \{\mathbf{y}_i\}_{i=1}^n$ , we want to find a model so that the probability of error is minimal when predicting  $\mathbf{y} \in Y$  for a new  $\mathbf{x} \in X$ . The goal is then to learn from the dataset

$D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$  a prediction function  $\bar{f}$  which maps image slices  $X$  to segment labels  $Y$ . In our setting a segment label  $\mathbf{y}$  is composed of real-valued threshold variables. This makes the function  $\bar{f}$  a regression model.  $\bar{f}$  belongs to a hypothesis space  $H = \{f(\cdot) | f : X \rightarrow Y\}$  and is evaluated to minimize a loss function  $L$  which measures the similarity between two outputs.

$$\bar{f} = \arg \min_{f \in H} L(f, D) = \arg \min_{f \in H} \sum_{i=1}^n L(\mathbf{y}_i, f(\mathbf{x}_i)),$$

where  $L$  denotes the error or loss of  $f$  on example  $(\mathbf{x}_i, \mathbf{y}_i)$  and  $L(f, D)$  denotes the collective loss of  $f$  on the training set  $D$ . We expect the loss to be close to zero when a pattern is detected, as it measures the discrepancy between the output of the prediction function and the correct output value. When a new image slice is presented, the target output is not available and then  $\bar{f}$  is used to predict  $\mathbf{y}$  for the given input  $\mathbf{x}$ . This implies a multi-target setting, however, in this work, we consider each  $y_i \in \mathbf{y}$  an independent target where  $y_i \in \mathbb{R}$ . We build the multi-target regressor by learning multiple independent regression functions, one for each target. A natural future investigation is whether a multi-target or *structured prediction* setting, in which dependencies between the multiple targets are considered during training/testing (i.e. the level of intensity can influence the saturation prediction), can improve results.

### 3.3 Regression trees

One way of defining the regression model is as a linear function where a real-valued dependent variable  $y_i$  is modeled as a linear function of a real-valued independent variable  $\mathbf{x}$  plus noise. In linear regression  $f_i$  is a global model, where there is a single predictive formula holding over the entire data-space. However, when the data is high-dimensional with non-sparse features which interact in complicated, nonlinear ways, assembling a single global model is not the best approach. One way to consider nonlinear regression is to partition the space into smaller regions which can be divided again until it fits the training data.

Regression trees use the tree to represent such recursive partitions and predict real-valued outcomes. Each of the terminal nodes (or leaves) of the tree represents a cell of the partition, and has attached a constant estimate of  $y$  which applies in that leaf only. That is, given the points  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_c, y_c)$  are the samples belonging to the leaf node  $l$ , the model for  $l$  is  $\hat{y} = \frac{1}{c} \sum_{i=1}^c y_i$  (or the sample mean of  $y_i$  in the leaf). A point  $\mathbf{x}$  belongs to a leaf if  $\mathbf{x}$  falls in the corresponding region of the partition. In our defined problem (as in Section 3.2), threshold values are predicted by traversing the regression tree until a leaf node is encountered, and the leaf outcome value  $\hat{y}$  is assigned to the unseen instance. The interior nodes are labeled with tests, and the edges or branches between them labeled with the answers. We start at the root node of the tree and apply a sequence of tests in the tree about the features to figure out the prediction leaf. Which test is performed next depends on the answers to previous questions. Figure 2 shows a regression tree for muscle nuclei detection learned with our framework. The inner nodes in the tree represent tests on certain feature values, while the leaf contains the outcome. The outcome is the real-valued intensity threshold needed to segment the image slice. In this case each test refers to only a single attribute, and has a yes or no answer, e.g., “Is Autocorrelation < 25.01 ?”.

Regression trees offer several advantages when compared to alternative methods (i.e., SVMs [3]): (i) fast algorithms exist to learn the trees and make predictions; this minimizes delays during interaction with the medical expert; (ii) the obtained models are interpretable which might help in gaining medical insight of the distinguishing properties of the identified regions; (iii) deals well with missing data; as even though a path to a prediction leaf might be unreachable, still a prediction can be made by aggregating predictions of all leaves in the reachable sub-tree. This property could be exploited to still provide predictions in severely damaged tissue samples for which not all features can be evaluated; (iv) the model obtained gives a ridged response, so it can work when the true regression surface is not smooth.

**Learning** We start building the tree by finding the one binary test which maximizes the information we get about  $y$ . One commonly used *splitting criterion* in regression is the sum of squared errors, also employed by our work; this gives us our root node and two children nodes. At each child node, we repeat the same procedure. The sum of squared errors for a tree  $T$  is

$$S = \sum_{l \in \text{leaves}(T)} \sum_{i \in l} (y_i - \hat{y}_l)^2,$$

Feature	Description
<i>Haralick features</i>	based on the gray-level co-occurrence matrix (GLCM) of an image
<i>Total coarseness</i>	refers to notable variations of the grey levels
<i>Coarseness histogram</i>	characterizes the distribution of the coarseness
<i>Contrast</i>	the range of the pixel intensities
<i>Directionality</i>	indicates whether the image favors a certain direction
<i>Homogeneity</i>	closeness of the GLCM elements distribution to the GLCM diagonal
<i>Gray level statistics</i>	refers to the pixels gray-level values such as the average, the median, standard deviation, minimum and maximum values
<i>Color moments</i>	characterize the distribution of the color in an image
<i>Intensity histogram</i>	shows the number of pixels in an image at different intensity values
<i>Hue histogram</i>	shows the number of pixels in an image at different hue values
<i>Saturation histogram</i>	shows the number of pixels in an image at different saturation values
<i>Morphological features</i>	include properties of the slice such as its width, height and polar coordinates relative to the centre of the full image

Table 1: Features used to characterize an image slice and their descriptions.

where  $\hat{y}_l$  is the prediction for leaf  $l$ . We can re-write this as  $S = \sum_{l \in \text{leaves}(T)} n_l \cdot V_l$ , where  $V_l$  is the within-leave variance of leaf  $l$  and  $n_l$  is the number of examples in  $l$ . The split is made so as to minimize  $S$ .

A typical *stopping criterion* is to stop growing the tree when further splits gives less than some minimal amount of extra information  $\delta$  (the decrease in  $S$  becomes less than some threshold  $\delta$ ). Using the notation introduced above, the loss  $L$  can be then defined as the mean squared error between the predictions of tree to the data in  $X$ , compared to the true responses  $Y$ :

$$L = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y})^2.$$

### 3.4 Features

For each image slice we extract a set of features summarized in Table 1. We consider Haralick and Tamura texture features, homogeneity, gray level statistics, color and morphological features. By concatenating all features together we obtain the final feature descriptor consisting of 1022 elements. We have used in all settings and problems considered the same feature descriptors. Haralick features [9] are based on the gray-level co-occurrence matrix (GLCM) of an image and consist of the following: angular second moment, contrast, autocorrelation, variance, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance and difference entropy. Tamura features [26] characterize different texture properties such as total coarseness, coarseness histogram, contrast, directionality. The choice of color features was determined by the fact that most histology images are stained so as to emphasize medical aspects of interest. Typically, depending on the problem, a specific staining procedure is used. Texture features play also an important role as there can be a high variance in the types of tissue considered.

### 3.5 Label calibration

In Section 3.2 we defined a segment label  $\mathbf{y}$  as being composed of real-valued threshold variables corresponding to the intensity, hue and saturation channels. Imposing one threshold for each channel would only allow the detection of two types of regions, i.e., background and foreground. However, using targets such as lower and upper threshold values for all channels allows for a better segmentation. In this case every pixel with a value in one channel smaller than the lower threshold and larger than the upper threshold is removed. The combination of the three channels with both lower and upper threshold values as targets is a flexible and general enough to allow the segmentation of several types of histological tissue.

Thus, each slice is interactively labeled by the user with a target of form  $\mathbf{y} = (y_{il}, y_{hl}, y_{sl}, y_{iu}, y_{hu}, y_{su})$ , indicating lower and upper threshold bounds for the intensity, hue and saturation. One possible problem could be the introduction of inconsistencies in the regression model. This is due to the fact that the upper and lower thresholds can be widely spread, producing the same image segmentation for multiple values.

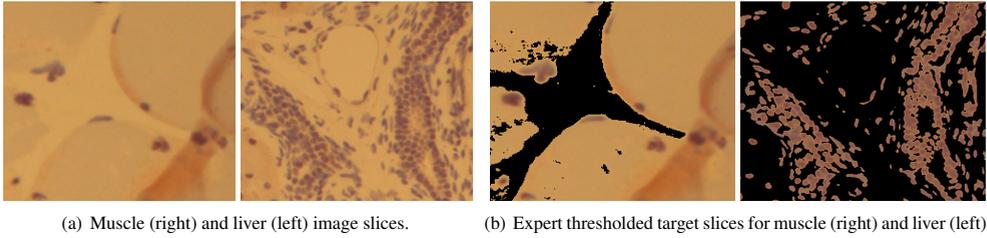


Figure 3: Examples of histology image slices and their expert user targets.

Segmenting the same image slice using two different set of target thresholds can produce indistinguishable segmentations from the medical point of view. To address this issue, we introduce a calibration phase, which separates the lower and upper thresholds for each channel as much as possible. This is done by increasing (decreasing) the lower (upper) values until a segmentation using these new values produces a slice that is different enough from the original slice. We consider that an image slice  $s_1$  is *similar* to a slice  $s_2$  if  $\frac{\# \text{ removed pixels } s_2}{\# \text{ removed pixels } s_1} < \varepsilon$ , where  $\varepsilon$  represents an upper bound on the difference in terms of the number of pixels between  $s_1$  and  $s_2$ , such that the two slices can be considered similar enough. Practically, we found that  $\varepsilon = 1.05$  is a reasonable value. Note that by increasing (decreasing) the lower (upper) values, it is only possible to obtain more, and not less, black pixels than the original window. The slice with most pixels removed will always appear in the numerator of the fraction.

## 4 Experiments

To illustrate the performance of our algorithm we report experiments with histology images of two tissue types. These images were obtained by medical experts from the intensive care research group of the KU Leuven, as part of their routine analysis of critical illness and response to therapy at the cellular level. The images pertain to liver and skeletal muscle tissue. In each case, we consider a different problem, that is the segmentation of a different region of interest. For both types of tissues the staining used emphasizes ubiquitinated protein accumulation. All images were taken at a 10-fold magnification factor using a Leica DM3000 microscope. They contain two types of regions of interest: cell nuclei for liver and muscle and fibers for muscle tissue. Examples of image slices for both tissues are illustrated in Figure 3. For liver we consider the problem of nuclei detection, while for muscle the learning has as target nuclei and fiber detection and the exclusion of the overlap-artifacts. Figure 3 shows the two learning problems.

To evaluate the performance of our method we build two datasets, one for liver and one for muscle. For muscle tissue, we used in total 15 full images, out of which 5 for training and 10 for testing. To obtain training and testing examples we randomly sampled 133 training instances and 136 testing instances. For liver tissue, we used 11 full images, out of which 5 for training and 6 for testing. We randomly sampled 99 instances for training and 105 instances for testing. Each image slice was manually annotated by the expert via the GUI's slider, with a set of targets which produced the best medical segmentation. To compare the prediction made for a slice with the ground truth annotation of the same slice, we use *accuracy* as measure. It calculates the ratio of the amount of pixels that match to the total amount of pixels from the two slices.

$$Acc_s = \frac{\#(c_{ij} = 0)}{m \times n} \times 100,$$

where  $c_{ij} = a_{ij} - b_{ij}$  with  $a_{ij}$  and  $b_{ij}$ , elements on the  $i$ -th row and  $j$ -th column of an image slice  $s$  and its ground truth, respectively;  $m \times n$  is the size of the slice. The final accuracy is the average over all test slices.

Table 2 shows results for liver and muscle datasets. We report result for the settings when we use calibrated labels and when the calibration was not considered. As a baseline we considered the global image segmentation using thresholding done by the expert. This was done without employing our model. Our algorithm significantly outperforms the global baseline by 43% for the muscle and 12% for liver. We also see that label calibration has a positive influence on the result. To select the parameters of the model (i.e.  $\delta$  and minimum number of examples to perform the split), we performed 10-fold internal cross-validation on the training set for both types of tissue.

	Method	Calibration	Result
<b>Liver</b>	L	Yes	<b>93.3%</b>
		No	91.3%
	G		82.6%
<b>Muscle</b>	L	Yes	<b>87.4%</b>
		No	86.1%
	G		42.0%

Table 2: Experimental results on the liver and muscle datasets. L indicates our local method, G is the global baseline. The best result is marked with bold.

## 5 Conclusions and Future Work

We introduce a new framework which can segment regions of interest in histology images, independently of the problem and tissue type. By combining simple image processing with an interactive machine learning technique, we outperform global thresholding approaches. These are commonly used in medical imaging, yet are more suitable for a certain type of tissue or task. Additionally, our framework is able to learn to automatically segment histology images by interacting with medical experts via a GUI. In this way, medical experts can define multiple and different learning targets, independently of the tissue type.

As future work we plan to investigate whether formalizing the region detection as a structured prediction problem can improve current results. Another line of research is employing online learning approaches, where the model is built incrementally. Also interesting to investigate is whether other machine learning methods (e.g., linear regression, random forests, (multi-class) classification [24], etc) give better results. On the practical level, one useful feature to reduce the training data is to generate informed samples in an active learning setting instead of randomly sampling slices from the image. A relevance feedback or preference elicitation setting is also worth investigating.

## References

- [1] Features for histology images. <http://www.informed.unal.edu.co/jccaicedo/docs/review.pdf>, November 2009.
- [2] N. Bonnet, J. Cutrona, and M. Herbin. A 'no-threshold' histogram-based image segmentation method. *Pattern Recognition*, 35(10):2319–2322, October 2002.
- [3] Bernhard E. Boser and et al. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.
- [4] Thomas Brox, Yoo-Jin Kim, Joachim Weickert, and Wolfgang Feiden. Fully-automated analysis of muscle fiber images with combined region and edge-based active contours. In *Bildverarbeitung für die Medizin*, pages 86–90, 2006.
- [5] A.E. Carpenter, T.R. Jones, M.R. Lamprecht, C. Clarke, I.H. Kang, O. Friman, D.A. Guertin, J.H. Chang, R.A. Lindquist, J. Moffat, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):R100, 2006.
- [6] L. Chen and P. Pu. Survey of preference elicitation methods. *Swiss Federal Institute of Technology in Lausanne, Technical Report No. IC/200467*, 2004.
- [7] Marie Dumont, Raphaël Marée, Louis Wehenkel, and Pierre Geurts. Fast multi-class image annotation with random subwindows and multiple output randomized trees. In *VISAPP*, volume 2, pages 196–203. INSTICC, feb 2009.
- [8] M.N. Gurcan, L.E. Boucheron, A. Can, A. Madabhushi, N.M. Rajpoot, and B. Yener. Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering*, 2:147–171, 2009.
- [9] Robert M. Haralick, K. Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, November 1973.

- [10] Lei He, L. Rodney Long, Sameer Antani, and George R. Thoma. Computer assisted diagnosis in histopathology. *Sequence and Genome Analysis: Methods and Applications*, pages 271–287, 2011.
- [11] Steven C.H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. Semi-supervised svm batch mode active learning for image retrieval. In *CVPR*, pages 24–26, Alaska, US, June 2008.
- [12] A.J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, pages 2372–2379. Ieee, 2009.
- [13] L.C. Junqueira and J. Carneiro. *Basic Histology: Text & Atlas*. basic histology. McGraw-Hill, 2005.
- [14] J. A. Kiernan. *Histological and Histochemical Methods: Theory and Practice*. Cold Spring Harbor Laboratory Press, 4 edition, 2008.
- [15] Inc. Mayachitra. Bring state-of-art image informatics solutions to your desktop, 2006-2001.
- [16] S. E. Mills. *Histology for Pathologists*. Lippincott Williams & Wilkins, 3 edition, 2006.
- [17] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11:285–296, 1975.
- [18] N.R. Pal and S.K. Pal. A review on image segmentation techniques. *Pattern recognition*, 26(9):1277–1294, 1993.
- [19] Radu Rogojanu, Giovanna Bises, Cristian Smochina, and Vasile Manta. Segmentation of cell nuclei within complex configurations in images with colon sections. *International Conference on Intelligent Computer Communication and Processing*, 0:243–246, 2010.
- [20] Ross, M., Kaye, G. I., and W Pawlina. *Histology: a Text and Atlas*. Lippincott Williams & Wilkins, 4 edition, 2002.
- [21] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13:146, 2004.
- [22] H. Shen, G. Nelson, D.E. Nelson, S. Kennedy, D.G. Spiller, T. Griffiths, N. Paton, S.G. Oliver, M.R.H. White, and D.B. Kell. Automated tracking of gene expression in individual cells and cell compartments. *Journal of the Royal Society Interface*, 3(11):787, 2006.
- [23] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *TPAMI*, 22(12):1349–1380, December 2000.
- [24] C. Sommer, C. Straehle, U. Kothe, and F. A. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. pages 230–233, March 2011.
- [25] R. Szeliski. *Computer vision: Algorithms and applications*. Springer-Verlag New York Inc, 2010.
- [26] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on System, Man and Cybernatic*, 6, 1978.
- [27] Alison Todman, Ela Claridge, Alison G. Todman, and Ela Claridge. Cell segmentation in histological images of striated muscle tissue- a perceptual grouping approach, 2008.
- [28] Zuva Tranos, Olugbara Oludayo O., Ojo Sunday O., and Ngwira Seleman M. Image segmentation, available techniques, developments and open issues. *Canadian Journal on Image Processing and Computer Vision*, pages 20–29, 2011.
- [29] Ching-Wei Wang. A bayesian learning application to automated tumour segmentation for tissue microarray analysis. In *MLMI*, pages 100–107, Berlin, Heidelberg, 2010. Springer-Verlag.
- [30] Q. Wang, J. Niemi, C.M. Tan, L. You, and M. West. Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy. *Cytometry Part A*, 77(1):101–110, 2010.
- [31] Zhaozheng Yin, Ryoma Bise, Mei Chen, and Takeo Kanade. Cell segmentation in microscopy imagery using a bag of local bayesian classifiers. In *ISBI*, April 2010.
- [32] Xiang Sean Zhou and Thomas S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Syst.*, 8(6):536–544, 2003.

# Modeling and Evaluating Cooperation in Multi-Context Systems using Conviviality

Vasileios Efthymiou <sup>a</sup>

Patrice Caire <sup>a</sup>

Antonis Bikakis <sup>b</sup>

<sup>a</sup> *University of Luxembourg, Interdisciplinary Center for Security, Reliability and Trust*

<sup>b</sup> *Department of Information Studies, University College London*

## Abstract

Multi-Context Systems is a rule-based representation model for distributed, heterogeneous knowledge agents, which cooperate by sharing parts of their local knowledge through a set of bridge rules also known as mappings. The concept of conviviality was recently proposed for modeling and measuring cooperation among agents in multiagent systems. In this paper, we describe how conviviality can be used to model and evaluate cooperation in Multi-Context Systems. As a potential application, we also propose a conviviality-based method for inconsistency resolution based on the idea that the optimal solution is the one that minimally decreases the conviviality of the system.

## 1 Introduction

*Multi-Context Systems (MCS)* [1, 2, 3] are logical formalizations of distributed context theories connected through a set of bridge rules, which enable information flow between contexts. A *context* can be thought of as a logical theory - a set of axioms and inference rules - that models local knowledge. Intuitively, MCS can be used as a representation model for any information system that involves distributed, heterogeneous knowledge agents including peer-to-peer systems, distributed ontologies (e.g., Linked Open Data) or Ambient Intelligence systems. In fact, several applications have already been developed on top of MCS or other similar formal models of context including (a) the CYC common sense knowledge base [4]; (b) contextualized ontology languages, such as Distributed Description Logics [5] and C-OWL [6]; (c) context-based agent architectures [7, 8]; and (d) distributed reasoning algorithms for Mobile Social Networks [9] and Ambient Intelligence systems [10].

The individual entities that such systems consist of cooperate by sharing information through their bridge rules. By combining and reasoning on the information they import they are able to derive new knowledge. This feature is enabled by the notions of contexts, bridge rules and contextual reasoning used in MCS. But, how can we then evaluate the ways in which the system enables this cooperation? How can we characterise a MCS based on the opportunities for information exchange that it provides to its contexts? To answer such questions, we introduce in MCS the notion of *conviviality*.

Defined by Illich as “individual freedom realized in personal interdependence” [11], conviviality has been introduced as a social science concept for multiagent systems to highlight soft qualitative requirements like user friendliness of systems. Multiagent systems technology can be used to realize tools for conviviality when we interpret “freedom” as choice [12]. Tools for conviviality are concerned in particular with dynamic aspects of conviviality, such as the emergence of conviviality from the sharing of properties or behaviors whereby each member’s perception is that their personal needs are taken care of [11]. We measure conviviality by counting the possible ways to cooperate, indicating degree of choice or freedom to engage in coalitions. Our coalitional theory is based on dependence networks [13, 14], labeled directed graphs where the nodes are agents, and each labeled edge represents that the former agent depends on the latter one to achieve some goal. The focus on dependence networks and more specifically on their cycles, is a reasonable way of formalizing conviviality as something related to the freedom of choice of individuals plus the subsidiary relations –interdependence for task achievement– among fellow members of a social system.

In distributed information systems, individual freedom is linked with the choice to keep personal knowledge and beliefs at the local level, while interdependence is understood as reciprocity, i.e. cooperation. Participating entities depend on each other to achieve the enrichment of their local knowledge.

Considering the potential applications of MCS and the notion of conviviality as described above, our main research question is the following: *How to introduce the concept of conviviality to Multi-Context Systems?* This main research question breaks into the following questions: How to define and model conviviality for Multi-Context Systems? How to measure the conviviality of Multi-Context Systems? How to use conviviality as a property of Multi-Context Systems?

Building on the ideas of [15], where we identified ways in which conviviality tools, and specifically dependence networks and conviviality measures, can be used to evaluate cooperation in Contextual Defeasible Logic (which can be viewed as a specific case of MCS), we propose: *i.*) a formal model for representing *information dependencies* in MCS based on dependence networks, *ii.*) conviviality measures for MCS and *iii.*) a potential application of these tools for the problem of inconsistency resolution in MCS.

So far, most approaches for inconsistency resolution in MCS have been based on the *invalidation* or *unconditional application* of a subset of the bridge rules that cause inconsistency. They differ in the preference criterion that is applied for choosing among the candidate solutions. Here, we propose using the conviviality of the system as a preference criterion, based on the idea that removing (or applying unconditionally) a bridge rule affects the information dependency between the connected contexts, and, as a result, the conviviality of the system. We suggest that the optimal solution is the one that minimally affects conviviality.

The rest of the paper is structured as follows: Section 2 presents formal definitions for MCS, as these were originally proposed in [3]. Section 3 proposes a model and measures for conviviality in MCS. Section 4 describes a potential use of conviviality as a property of MCS for the problem of inconsistency resolution. Last section summarizes and presents directions for future work in the field.

## 2 Multi-Context Systems - Formal Definitions

For the needs of this paper we will use the definition of heterogeneous nonmonotonic MCS given in [3], according to which a MCS is a set of contexts, each composed of a knowledge base with an underlying logic, and a set of bridge rules which control the information flow between contexts. A logic  $L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$  consists of the following components:

- $\mathbf{KB}_L$  is the set of well-formed knowledge bases of  $L$ . Each element of  $\mathbf{KB}_L$  is a set of formulas.
- $\mathbf{BS}_L$  is the set of possible belief sets, where the elements of a belief set is a set of formulas.
- $\mathbf{ACC}_L: \mathbf{KB}_L \rightarrow 2\mathbf{BS}_L$  is a function describing the semantics of the logic by assigning to each knowledge base a set of acceptable belief sets.

A *bridge rule* can add information to a context, depending on the belief sets which are accepted at other contexts. Let  $L = (L_1, \dots, L_n)$  be a sequence of logics. An  $L_k$ -bridge rule  $r$  over  $L$  is of the form

$$r = (k : s) \leftarrow (c_1 : p_1), \dots, (c_j : p_j), \mathbf{not}(c_{j+1} : p_{j+1}), \dots, \mathbf{not}(c_m : p_m). \quad (1)$$

where  $1 \leq c_i \leq n$ ,  $p_i$  is an element of some belief set of  $L_{c_i}$ ,  $k$  refers to the context receiving information  $s$ . We denote by  $h_b(r)$  the belief formula  $s$  in the head of  $r$ .

A *MCS*  $M = (C_1, \dots, C_n)$  is a collection of contexts  $C_i = (L_i, kb_i, br_i)$ ,  $1 \leq c_i \leq n$ , where  $L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$  is a logic,  $kb_i \in \mathbf{KB}_i$  a knowledge base, and  $br_i$  a set of  $L_i$ -bridge rules over  $(L_1, \dots, L_n)$ . For each  $H \subseteq \{h_b(r) | r \in br_i\}$  it holds that  $kb_i \cup H \in \mathbf{KB}_{L_i}$ , meaning that bridge rule heads are compatible with knowledge bases.

A *belief state* of  $M = (C_1, \dots, C_n)$  is a sequence  $S = (S_1, \dots, S_n)$  such that  $S_i \in \mathbf{BS}_i$ . Intuitively,  $S$  is derived from the knowledge of each context and the information conveyed through applicable bridge rules. A bridge rule is applicable in a belief state  $S$  iff for  $1 \leq i \leq j$ :  $p_i \in S_{c_i}$  and for  $j < l \leq m$ :  $p_l \notin S_{c_l}$ . By  $br_M = \bigcup_{i=1}^n br_i$  we denote the set of all bridge rules of  $M$ .

**Example 1.** Consider a MCS  $M$ , through which the software agents of three research students exchange information and classify research articles that they access in online databases.  $M$  contains contexts  $C_1 - C_3$ , each encoding the knowledge of each of the three agents. The knowledge bases for the three contexts are:

$$\begin{aligned} kb_1 &= \{sensors, corba, centralizedComputing \leftrightarrow \neg distributedComputing\} \\ kb_2 &= \{profA\} \end{aligned}$$

$$kb_3 = \{ubiquitousComputing \sqsubseteq ambientComputing\}$$

$C_1$  collects information about the keywords of the articles and encodes this information in propositional logic. In this case, the article under examination is about sensors and corba (Common Object Request Broker Architecture).  $C_1$  also possesses the knowledge that centralized computing and distributed computing are two complementary concepts.  $C_2$  uses propositional logic to encode additional information about articles, including the names of their authors; in this case  $profA$  is the author of the article under examination. Finally,  $C_3$  is an ontology of computing-related concepts, according to which *ubiquitousComputing* is a type of *ambientComputing*.

The bridge rules that the three agents use to exchange information and collectively decide about the classification of the article are as follows:

$$\begin{aligned} r_1 &= (1 : centralizedComputing) \leftarrow (2 : middleware) \\ r_2 &= (1 : distributedComputing) \leftarrow (3 : ambientComputing) \\ r_3 &= (2 : middleware) \leftarrow (1 : corba) \\ r_4 &= (3 : ubiquitousComputing) \leftarrow (1 : sensors), (2 : profB) \end{aligned}$$

Rule  $r_1$  links the concept of *middleware* used by  $C_2$  to the concept of *centralized-Computing* of  $C_1$ .  $r_2$  expresses that *ambientComputing* (a term used by  $C_3$ ) implies *distributedComputing* (a term used by  $C_1$ ).  $r_3$  expresses that *corba* is a type of *middleware*, while  $r_4$  expresses the belief of the third agent ( $C_3$ ) that articles that are written by  $profB$  and that contain sensors among their keywords are about *ubiquitousComputing*.

Equilibrium semantics selects certain belief states of a MCS  $M = (C_1, \dots, C_n)$  as acceptable. Intuitively, an equilibrium is a belief state  $S = (S_1, \dots, S_n)$  where each context  $C_i$  respects all bridge rules applicable in  $S$  and accepts  $S_i$ . Formally,  $S$  is an equilibrium of  $M$ , iff for  $1 \leq i \leq n$ ,

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{h_b(r) \mid r \in br_i \text{ applicable in } S\}).$$

**Example 2.** In the example given above,  $S = (S_1, S_2, S_3)$  is the only equilibrium of the system:

$$S = (\{sensors, corba, centralizedComputing\}, \{profA, middleware\}, \emptyset).$$

$S_3$  is an empty set, since  $kb_3$ , the knowledge base of  $C_3$ , is an empty set,  $br_3 = \{r_4\}$ , namely the set of bridge rules for context  $C_3$  only consists of bridge rule  $r_4$ , and  $r_4$  is not applicable in  $S$ , because  $profB \notin S_2$ .

### 3 Modeling and measuring conviviality in MCS

We mentioned in the introduction that dependence networks have been proposed as a model for representing social dependencies among the agents of a multiagent system. It has also been used as the underlying model for formalizing and measuring conviviality in such systems. In this section, we describe how dependence networks can be used to model the information dependencies among the contexts of a MCS and how conviviality measures can then be applied in MCS. Our approach is based on the following ideas: (a) cooperation in MCS can be understood as information sharing among the contexts; (b) it is enabled by the bridge rules of the system; and (c) therefore, bridge rules actually represent information dependencies among the contexts. The more bridges between the contexts, the more possibilities for cooperation and information exchange. On the other hand, no bridge rules would mean that the different contexts are actually autonomous systems, which do not share their local knowledge.

#### 3.1 Dependence Networks Model for MCS

Conviviality may be modeled by the reciprocity-based coalitions that can be formed [16]. Some coalitions, however, provide more opportunities for their participants to cooperate than others, being thereby more convivial. To represent the interdependencies among agents in the coalitions, [16] use dependence networks.

In this subsection, we first present Definition 3.1 from [16], which abstracts from tasks and plans. Then, building on [16]'s definition, we introduce our definition for a dependence network corresponding to a MCS.

A dependence network is defined by [16] as follows:

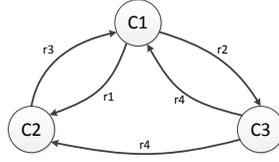


Figure 1: The dependence network  $DN(M)$  of MCS  $M$  of the running example.

**Definition 3.1** (Dependence networks). A *dependence network* (DN) is a tuple  $\langle A, G, dep, \geq \rangle$  where:  $A$  is a set of agents,  $G$  is a set of goals,  $dep : A \times A \rightarrow 2^G$  is a function that relates with each pair of agents, the sets of goals on which the first agent depends on the second, and  $\geq : A \rightarrow 2^G \times 2^G$  is for each agent a total pre-order on sets of goals occurring in its dependencies:  $G_1 >_{(a)} G_2$ .

To capture the notions of *contexts* and *bridge rules*, we now introduce our definition, Definition 3.2, for a dependence network corresponding to a MCS, as follows:

**Definition 3.2** (Dependence networks for MCS). A *dependence network corresponding to a MCS  $M$* , denoted as  $DN(M)$ , is a tuple  $\langle C, R, dep, \geq \rangle$  where:  $C$  is the set of contexts in  $M$ ;  $R$  is the set of bridge rules in  $M$ ;  $dep : C \times C \rightarrow 2^R$  is a function that is constructed as follows: for each bridge rule  $r$  (in the form of (1)) in  $R$  add the following dependencies:  $dep(k, c_i) = \{r\}$  where  $k$  is the context appearing in the head of  $r$  and  $c_i$  stands for each distinct context appearing in the body of  $r$ ; and  $\geq : C \rightarrow 2^R \times 2^R$  is for each context a total pre-order on sets of bridge rules that the context appears in their heads.

In other words, a bridge rule  $r$  creates one dependency between context  $k$ , which appears in the head of  $r$ , and each of contexts  $c_i$  that appear in the body of  $r$ . The intuition behind this is that  $k$  depends on the information it receives from each  $c_i$  to achieve its goal, which is to apply  $r$  in order to infer  $s$ .

We should also note here that the total preorder that each context defines on the sets of bridge rules may reflect the local preferences of a context, e.g., in the way that these are defined and used in Contextual Defeasible Logic [17, 10]. For sake of simplicity, we do not use this feature in the conviviality model that we describe below. However, it is among our plans to integrate it in future extensions of this work.

To graphically represent dependence networks, we use nodes for contexts and labeled arrows for dependencies among the contexts that the arrows connect. An arrow from context  $a$  to context  $b$ , labeled as  $r$ , means that  $a$  depends on  $b$  to apply rule  $r$ . In our running example, the dependence network that corresponds to MCS  $M$  is the one depicted in Figure 1.

In this graph, each node corresponds to one of the contexts in  $M$ . Dependencies are derived from the four bridge rules of  $M$ . For example, there are two dependencies labeled by  $r_4$ : each of them connects  $C_3$ , which appears in the head of  $r_4$ , to one of the contexts  $C_1$  and  $C_2$ , which appear in the body of  $r_4$ . This actually means that to apply rule  $r_4$  in order to prove that the paper under examination is about ubiquitous computing,  $C_4$  depends on information about the keywords of the paper that it imports from  $C_1$  and information about the authors of the paper that it imports from  $C_2$ .

## 3.2 Conviviality Measures

Conviviality measures have been introduced to compare the conviviality of multi agent systems [16], for example before and after, making a change such as adding a new norm, or policy. Furthermore, to evaluate conviviality in a more precise way, [16] introduce formal conviviality measures for dependence networks using coalitional game theoretic framework. Based on Illich’s definition of conviviality as “individual freedom realized in personal interdependency”, the notions of interdependency and choice, if we interpret freedom as choice, are stressed. Such measures provide insights into the type of properties that may be measured in convivial systems and thus reveal the quality of the system.

The conviviality measures presented in this work reflect the following Assumptions:

- A1 the cycles identified in a dependence network are considered as coalitions. These coalitions are used to evaluate conviviality in the network. Cycles are the smallest graph topology expressing interdependence, thereby conviviality, and are therefore considered atomic relations of interdependence. When referring to *cycles*, we are implicitly signifying *simple cycles*, i.e., where all nodes are distinct [18]; we also discard self-loops. When referring to conviviality, we always refer to potential interaction not actual interaction.

A2 conviviality in a dependence network is evaluated in a bounded domain, i.e., over a  $[min, max]$  interval. This allows the comparison of different systems in terms of conviviality.

A3 there is more conviviality in larger coalitions than in smaller ones.

A4 the more coalitions in the dependence network, the higher the conviviality measure (*ceteris paribus*).

Our top goal is to maximize conviviality in MCS. Some coalitions provide more opportunities for their participating contexts to cooperate than others, being thereby more convivial. Our Requirements are thus:

R1 maximize the size of the coalitions, i.e., maximize the number of contexts involved in the coalitions,

R2 maximize the number of these coalitions.

Following the definition of the *conviviality of a dependence network* [16], we define the *conviviality of a dependence network of a MCS M* as

$$\text{Conv}(DN(M)) = \frac{\sum_{c_i, c_j \in C, i \neq j} \text{coal}(c_i, c_j)}{\Omega}, \quad (2)$$

$$\Omega = |C|(|C| - 1) \times \Theta, \quad (3)$$

$$\Theta = \sum_{L=|C|}^{L=2} P(|C| - 2, L - 2) \times |R|^L, \quad (4)$$

where  $|C|$  is the number of contexts in  $M$ ,  $|R|$  is the number of bridge rules in  $M$ ,  $L$  is the cycle length,  $P$  is the usual permutation defined in combinatorics,  $\text{coal}(c_i, c_j)$  for any distinct  $c_i, c_j \in C$  is the number of cycles that contain both  $c_i$  and  $c_j$  in  $DN(M)$  and  $\Omega$  denotes the maximal number of pairs of contexts in cycles (which produces the normalization mentioned in Assumption A2).

This way, the conviviality measurement of a dependence network, which is a rational number in  $[0,1]$ , can be used to compare different dependence networks, with 0 being the conviviality of an acyclic dependence network and 1 the conviviality of a fully-connected dependence network.

**Example 3.** Following Equation 2 and the dependence network of  $M$ , which is graphically represented in Figure 1, we calculate the conviviality of  $DN(M)$  of our running example, as:

$$\text{Conv}(DN(M)) = \frac{10}{\Omega} = 0.208, \text{ where } \Omega = 480.$$

The result of Example 3 is just a way of comparing the conviviality of different systems. By itself it cannot be used to classify the conviviality of a MCS.

## 4 Use of conviviality as a property of MCS: Inconsistency Resolution

As we previously argued, conviviality is a property that characterizes the cooperativeness of a MCS, namely the alternative ways in which the contexts of a MCS can share information in order to derive new knowledge. By evaluating conviviality, the system may propose the different ways in which it can be increased, e.g., by suggesting new connections (bridge rules) between the system contexts.

Consider, for example, a MCS in which a context does not import any information from other contexts. Recommending other contexts that this context could import information from, could increase the conviviality of the system, which would in turn lead to enriching the local knowledge of the context, but also the knowledge of the whole system.

### 4.1 Problem Description

A potential way of using conviviality as a property of MCS, which we describe in more detail in this section, is for the problem of inconsistency resolution. In a MCS, even if contexts are locally consistent, their bridge rules may render the whole system inconsistent. This is formally described in [3] as a *lack of an equilibrium*. All techniques that have been proposed so far for inconsistency resolution are based on the same intuition: a subset of the bridge rules that cause inconsistency must be invalidated and another subset

must be unconditionally applied, so that the entire system becomes consistent again. For nonmonotonic MCS, this has been formally defined in [19] as diagnosis:

“Given a MCS  $M$ , a *diagnosis* of  $M$  is a pair  $(D_1, D_2)$ ,  $D_1, D_2 \subseteq br_M$ , s.t.  $M[br_M \setminus D_1 \cup heads(D_2)] \not\models \perp$ ”.  $D^\pm(M)$  is the set of all such diagnoses, while  $M[br_M \setminus D_1 \cup heads(D_2)]$  is the MCS obtained from  $M$  by removing the rules in  $D_1$  and adding the heads of the rules in  $D_2$ .

In other words, if we deactivate the rules in  $D_1$  and apply the rules in  $D_2$  in unconditional form,  $M$  will become consistent. In a MCS it is possible that there is more than one diagnosis that can be applied to restore consistency.

**Example 4.** In our running example, consider the case that *profB* is also identified by  $C_2$  as one of the authors of the paper under examination. In this case  $kb_2$  would also contain *profB*:

$$kb_2 = \{profA, profB\}$$

This addition would result in an inconsistency in  $kb_1$ , caused by the activation of rules  $r_4$  and  $r_2$ . Specifically, rule  $r_4$  would become applicable, *ubiquitousComputing* and *ambientComputing* would become true in  $C_3$ ,  $r_2$  would then become applicable too, and *distributedComputing* would become true in  $C_1$  causing an inconsistency with *centralizedComputing*, which has also been evaluated as true. To resolve this conflict, one of the four bridge rules  $r_1$ - $r_4$  must be invalidated. Using the definition of diagnosis that we presented above, this is formally described as:

$$D^\pm(M) = \{(\{r_1\}, \emptyset), (\{r_2\}, \emptyset), (\{r_3\}, \emptyset), (\{r_4\}, \emptyset)\}.$$

Various criteria have been proposed for choosing a diagnosis including: *i.*) the number of bridge rules contained in the diagnosis - specifically in [19] subset-minimal diagnoses are preferred, *ii.*) local preferences on diagnoses proposed in [20] and *iii.*) local preferences on contexts and provenance information, which have been proposed for Contextual Defeasible Logic [17, 10].

## 4.2 Proposed Solution

Our approach is to use the conviviality of the resulted system as a criterion for choosing a diagnosis. This actually means that for each candidate solution (diagnosis), we measure the conviviality of the system that is derived after applying the diagnosis, and we choose the diagnosis that minimally decreases the conviviality of the system. The intuition behind this approach is that the system should remain as *cooperative* as possible, and this is achieved by enabling the maximum number of agents to both *contribute* to and *benefit* from this cooperation. In the extreme case of removing all bridge rules, there will be no inconsistencies; however contexts will not be able to exchange information. Our proposed solution is to resolve inconsistencies, by also keeping as many bridge rules (hence possibilities for information exchange) as possible.

Diagnoses contain two types of changes applicable in the bridge rules: invalidation (removal) of a rule; and applying a rule unconditionally, which means removing the body of the rule. These changes affect the dependencies of the system as follows: When invalidating or adding unconditionally rule  $r$  (as defined in (1)) in a MCS  $M$ , all the dependencies labeled by  $r$  are removed from the dependence network of  $M$ .

Assuming that  $DN(M, D_i)$  is the dependence network that corresponds to MCS  $M$  after applying diagnosis  $D_i$ , the optimal diagnosis is the one that maximizes the conviviality of  $DN(M, D_i)$ :

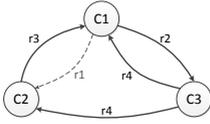
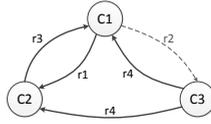
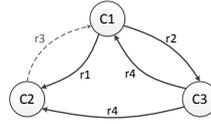
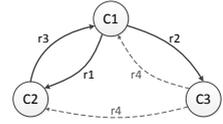
$$D_{opt} = \{D_i : Conv(DN(M, D_i)) = max\}$$

**Example 5.** In the running example, there are four diagnoses that we can choose from:  $D_1$ - $D_4$ . Each of them requires invalidating one of the four bridge rules  $r_1$  to  $r_4$ , respectively. Figures 2 to 5 depict the four dependence networks  $DN(M, D_i)$ , which are derived after applying  $D_i$ . Dashed arrows in Figures 2-5 represent the dependencies that are dropped in each  $DN(M, D_i)$ , by applying diagnosis  $D_i$ .

Following Equation 2 and the four dependence networks, which are graphically represented in Figures 2-5, the conviviality of each  $DN(M, D_i)$  is:

$$\begin{aligned} Conv(DN(M, D_1)) &= \frac{8}{\Omega} = 0.037 \text{ and} \\ Conv(DN(M, D_2)) &= Conv(DN(M, D_3)) = Conv(DN(M, D_4)) = \frac{2}{\Omega} = 0.009, \end{aligned}$$

with  $\Omega = 216$ .

Figure 2:  $DN(M, D_1)$ Figure 3:  $DN(M, D_2)$ Figure 4:  $DN(M, D_3)$ Figure 5:  $DN(M, D_4)$ 

Since the number of bridge rules  $|R|$  is now 3, instead of 4,  $\Omega$  has a different value than in  $DN(M)$ . By applying  $D_1$  (Figure 2), only one cycle  $(C_1, C_2)$  is removed from the initial dependence network  $DN(M)$ , illustrated in Figure 1. However, by applying any of the rest diagnoses  $D_2$ - $D_4$ , two cycles are removed from  $DN(M)$ . Specifically, by applying  $D_2$  (Figure 3), we remove the cycles  $(C_1, C_3)$  and  $(C_1, C_3, C_2)$ . By applying  $D_3$  (Figure 4), we remove the cycles  $(C_1, C_2)$  and  $(C_1, C_3, C_2)$ . Finally, by applying  $D_4$  (Figure 5), we remove the cycles  $(C_1, C_3)$  and  $(C_1, C_3, C_2)$ .

Therefore the optimal diagnosis is  $D_1$ . By applying  $D_1$  the system will have the equilibrium  $S'$ :

$$S' = (\{sensors, corba, distributedComputing\}, \{profA, profB, middleware\}, \{ubiquitousComputing, ambientComputing\})$$

This approach can also be combined with any of the approaches that have been proposed so far for inconsistency resolution. For example, one may choose to apply the conviviality-based approach only to those diagnoses that comply with some constraints representing user-defined criteria, as suggested by in [20]. It can also be combined with preferences on diagnoses proposed by [20] or preferences on contexts suggested by [17, 10]. A study of such combined approaches will be part of our future work in the field.

## 5 Conclusion

Today, with the rise of systems in which knowledge is distributed in a network of interconnected heterogeneous and evolving knowledge resources, such as Semantic Web, Linked Open Data, and Ambient Intelligence, research in contextual knowledge representation and reasoning has become particularly relevant. Multi-Context Systems (MCS) are logical formalizations of distributed context theories connected through a set of bridge rules, which enable information flow between contexts. The individual entities, that such systems consist of, cooperate by sharing information through their bridge rules. By combining and reasoning on the information they import, they are able to derive new knowledge. Evaluating the ways in which the system enables cooperations, and characterizing a MCS based on the opportunities for information exchange that it provides to its contexts are therefore, key issues. The social science concept of conviviality has recently been proposed to model and measure the potential cooperation among agents in multiagent systems and ambient intelligence systems. Furthermore, formal conviviality measures for dependence networks using a coalitional game theoretic framework, have been introduced. Roughly, more opportunities to work with other agents increase the conviviality of the system.

This paper is a preliminary step toward extending the concept of conviviality, modelled with dependence networks, to MCSs. First, we describe how conviviality can be used to model cooperation in MCS. Based on the intuition that contexts depend on the information they receive from other contexts to achieve their goals, i.e., apply specific bridge rules to infer particular information, we define dependence networks for MCS. Furthermore, the aim is for MCSs to be as cooperative as possible, and for contexts to have as many choices as possible to cooperate with other contexts. This results in MCS being as convivial as possible. In order to evaluate the conviviality of a MCS, we apply pairwise conviviality measures and allow for comparisons among MCS. Finally we propose a potential use of conviviality as a property of MCS for the problem of inconsistency resolution. Indeed, without considering contextual information, reasoning can easily run to inconsistency problems, for example, when considering knowledge in the wrong context. We propose a solution based on the idea that the optimal solution is the one that minimally decreases the conviviality of the system. We illustrate how to model, measure and use conviviality for MCS with a running example.

## Acknowledgement

Thanks to: National Research Fund, Luxembourg (I2R-SER-PFN-11COPA) CoPAInS project

## References

- [1] Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics, or: how we can do without modal logics. *Artificial Intelligence* **65**(1) (1994)
- [2] Ghidini, C., Giunchiglia, F.: Local Models Semantics, or contextual reasoning=locality+compatibility. *Artificial Intelligence* **127**(2) (2001) 221–259
- [3] Brewka, G., Roelofsen, F., Serafini, L.: Contextual Default Reasoning. In: *IJCAI*. (2007) 268–273
- [4] Lenat, D.B., Guha, R.V.: *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989)
- [5] Borgida, A., Serafini, L.: Distributed Description Logics: Assimilating Information from Peer Sources. *Journal of Data Semantics* **1** (2003) 153–184
- [6] Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing Ontologies. In: *International Semantic Web Conference*. (2003) 164–179
- [7] Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. *Journal of Logic and Computation* **8**(3) (1998) 261–292
- [8] Sabater, J., Sierra, C., Parsons, S., Jennings, N.R.: Engineering Executable Agents using Multi-context Systems. *Journal of Logic and Computation* **12**(3) (2002) 413–442
- [9] Antoniou, G., Papatheodorou, C., Bikakis, A.: Reasoning about Context in Ambient Intelligence Environments: A Report from the Field. In: *KR, AAAI Press* (2010) 557–559
- [10] Bikakis, A., Antoniou, G., Hassapis, P.: Strategies for contextual reasoning with conflicts in Ambient Intelligence. *Knowledge and Information Systems* **27**(1) (2011) 45–84
- [11] Illich, I.: *Tools for Conviviality*. Marion Boyars Publishers, London (1974)
- [12] Caire, P., Villata, S., Boella, G., van der Torre, L.: Conviviality masks in multiagent systems. In: *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Estoril, Portugal, May 12-16, 2008, Volume 3. (2008) 1265–1268
- [13] Castelfranchi, C.: The micro-macro constitution of power. *Protosociology* **18** (2003) 208–269
- [14] Sichman, J.S., Conte, R.: Multi-agent dependence by dependence graphs. In: *Procs. of The First Int. Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002*, ACM (2002) 483–490
- [15] Caire, P., Bikakis, A.: Enhancing Cooperation in Distributed Information Systems Using Conviviality and Multi-Context Systems. In: *Multi-disciplinary Trends in Artificial Intelligence - 5th International Workshop, MIWAI 2011, Hyderabad, India, December 7-9, 2011. Proceedings*. Volume 7080 of *Lecture Notes in Computer Science.*, Springer (2011) 14–25
- [16] Caire, P., Alcade, B., van der Torre, L., Sombattheera, C.: Conviviality measures. In: *10th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Taipei, Taiwan, May 2-6, 2011. (2011)
- [17] Bikakis, A., Antoniou, G.: Defeasible Contextual Reasoning with Arguments in Ambient Intelligence. *IEEE Trans. on Knowledge and Data Engineering* **22**(11) (2010) 1492–1506
- [18] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*. 2nd edn. The MIT Press (2001)
- [19] Eiter, T., Fink, M., Schüller, P., Weinzierl, A.: Finding Explanations of Inconsistency in Multi-Context Systems. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*, AAAI Press (2010)
- [20] Eiter, T., Fink, M., Weinzierl, A.: Preference-Based Inconsistency Assessment in Multi-Context Systems. In: *Logics in Artificial Intelligence - 12th European Conference, JELIA 2010, Helsinki, Finland, September 13-15, 2010. Proceedings*. Volume 6341 of *Lecture Notes in Computer Science.*, Springer (2010) 143–155

# Automated Analysis of Social Choice Problems: Approval Elections with Small Fields of Candidates

Ulle Endriss  
Institute for Logic, Language and Computation  
University of Amsterdam

## Abstract

We analyse the incentives of a voter to vote insincerely in an election conducted under the system of approval voting. Central to our analysis are the assumptions we make on how voters deal with the uncertainty stemming from the fact that a tie-breaking rule may have to be invoked to determine the unique election winner. Because we only make very weak assumptions in this respect, it is impossible to obtain general positive results. Instead, we conduct a fine-grained analysis using an automated approach that reveals a clear picture of the precise conditions under which insincere voting can be ruled out. At the methodological level, this approach complements other recent work involving the application of techniques originating in computer science and artificial intelligence in the domain of social choice theory.

## 1 Introduction

Voting theory and, more generally speaking, social choice theory, originated in economics and political science, but are becoming increasingly important for artificial intelligence (AI). There are two distinct reasons for this trend. First, as the formal study of collective decision making, social choice theory has the potential to make important contributions to the design and analysis of multiagent systems. Second, methods of computer science and AI have turned out to be very helpful in analysing problems of social choice, leading to the interdisciplinary research area known as *computational social choice* (Chevalerey et al., 2007; Brandt et al., 2012). Maybe the clearest example for this kind of research is the large body of work devoted to the use of complexity theory for the analysis of the manipulation problem in voting, which has recently been reviewed by Faliszewski and Procaccia (2010). Other examples include the design of fast algorithms for computing the winners under complex voting rules (see, e.g., Conitzer et al., 2006) and the study of social choice problems with very large numbers of alternatives induced by the multi-attribute structure of decision problems arising in practice (see, e.g. Lang, 2004).

A collection of techniques from AI that has great potential for social choice theory but that has only received little attention to date is *automated reasoning*. It might be used to verify existing theorems in social choice theory and to search for new ones; it might be used to automatically check whether a given social choice rule meets a set of requirements; and it might assist in the design of new rules. So far, most efforts along this line have concentrated on one of the classical results in the field, Arrow's Theorem, which establishes the impossibility of devising a method for preference aggregation that simultaneously meets a small number of seemingly innocent requirements (Arrow, 1963). There have been attempts to prove (or at least verify proofs of) Arrow's Theorem (or special cases of Arrow's Theorem) using higher-order theorem provers (Nipkow, 2009; Wiedijk, 2007), first-order theorem provers (Grandi and Endriss, 2012), and SAT-solvers (Tang and Lin, 2009). One of the very few examples of using automated reasoning in other areas of computational social choice is our recent work on using a SAT-solver to automatically search for impossibility theorems in the domain of ranking sets of objects (Geist and Endriss, 2011).

In this paper we focus on yet another problem domain and we explore a much simpler approach.

### 1.1 The Problem: Sincerity and Manipulation in Approval Voting

One of the central questions in voting theory is under what circumstances a voter will have an incentive to *manipulate* the election by misrepresenting her true preferences. For instance, under plurality voting (where

each voter can award 1 point to one and only one candidate), if a voter favours a candidate from a small party who has no realistic chance of winning, then she has an incentive to manipulate and vote for her most preferred candidate from a mainstream party instead. The classical Gibbard-Satterthwaite Theorem shows that this problem cannot be avoided in general for any voting rule under which voters report a (possibly truncated) ranking of candidates (Taylor, 2005).

Under the system of *approval voting* (AV), a voter can approve of as many candidates as she likes and the candidate with the most approvals wins (Brams and Fishburn, 1978). That is, under AV voters report *sets* of candidates rather than *rankings*, which means that the Gibbard-Satterthwaite Theorem does not apply directly. In fact, if we continue to assume that true preferences are rankings of candidates, then the notion of truthful voting ceases to be well-defined (if you must report a set rather than a ranking, you certainly cannot report your truthful ranking). There is however a weaker notion of “good” behaviour in AV: a ballot  $B$  (a set of approved candidates) is called *sincere* if our voter does not prefer any of the candidates outside of  $B$  to any of those inside of  $B$ . We shall be interested in the following question: Assuming that a voter has obtained complete information on how all other voters are going to vote, under what circumstances can we ensure that she will never have an incentive to vote by means of an insincere ballot?

As it turns out, the answer to this question crucially depends on the assumptions we are willing to make on how a voter deals with the uncertainty arising from the fact that her vote might result in a tied election, meaning that the eventual winner must be chosen from a set of front-runners using a suitable tie-breaking rule. That is, when contemplating her ballot, the voter actually has to choose between several possible sets of front-runners rather than between several unique winners. On the other hand, her preferences are initially only defined over individual candidates. Thus, we need to reason about how a voter will extend her preferences to sets of candidates. If we make relatively strong assumptions regarding this matter, then we can obtain strong positive results showing that a fully informed voter will never have any cause to vote insincerely (Endriss, 2012). But these results break down when we weaken these assumptions. Our goal in this paper is to understand the case of weak assumptions regarding preference extension in more depth.

## 1.2 The Approach: Automated Analysis for a Fixed Number of Candidates

A simple but crucial insight is that we can abstract away from the number of voters. All that matters for the analysis of a manipulation situation is which candidates have obtained the most approvals, which are trailing by only 1 point each, and which are lagging behind even further (and thus certainly will not win). This means that it is possible, at least in principle, to exhaustively enumerate and check all relevant situations that could possibly arise, for any given number  $n$  of candidates. In this paper, we show that for small values of  $n$  this approach is also feasible in practice and that it allows us to derive interesting new results clarifying the precise conditions under which insincere voting might occur under AV.

The remainder of this paper is organised as follows. Section 2 introduces three principles for preference extension and sketches an algorithm for deciding whether one set of candidates dominates another under such a principle. Section 3 explains our approach to the automated analysis of the sincerity problem in AV with a fixed number of candidates and presents the results obtained. Section 4 concludes.

## 2 Preference Extensions

Let  $\mathcal{X}$  be a finite set of two or more *candidates* and let  $n := |\mathcal{X}|$ . Let  $\underline{\mathcal{X}} = 2^{\mathcal{X}} \setminus \{\emptyset\}$  denote the set of *nonempty sets of candidates*. We will consider a *voter* with a *preference relation*  $\succcurlyeq$  over *individual candidates* that is a *total order* (i.e., a binary relation on  $\mathcal{X}$  that is reflexive, transitive, complete and antisymmetric). That is, we write  $a \succcurlyeq b$  to express that our voter likes candidate  $a$  at least as much as  $b$ . The corresponding strict preference relation is denoted by  $\succ$ ; i.e.,  $a \succ b$  if  $a \succcurlyeq b$  but not  $b \succcurlyeq a$ .

Suppose our voter can choose between two alternative outcomes of an election,  $A$  and  $B$ , but for either one of them two or more candidates might be tied for winning; i.e.,  $A$  and  $B$  are elements of  $\underline{\mathcal{X}}$  and a unique winner would eventually have to be chosen from  $A$  or  $B$  using a tie-breaking rule (e.g., choosing randomly or asking an arbiter). Can we predict whether our voter will prefer  $A$  or  $B$ , given that we only know  $\succcurlyeq$ ?

In general, we cannot. But there are some reasonable assumptions that we can make. We will now introduce three principles for extending a preference order  $\succcurlyeq$  on  $\mathcal{X}$  to a preference order  $\succcurlyeq$  on  $\underline{\mathcal{X}}$ . Here,  $\succcurlyeq$  is assumed to be a *weak order* (i.e., a binary relation that is reflexive, transitive and complete). That is, while we assume that our voter can strictly rank all individual candidates, she might be indifferent between two sets of candidates. We write  $\succ$  for the strict part of  $\succcurlyeq$ . This form of preference extension has been studied

extensively under the name of *ranking sets of objects* (see, e.g., Barberà et al., 2004; Geist and Endriss, 2011). Each of the three principles we present will (usually) not be sufficient to allow us to induce the full weak order  $\succsim$ , but it will allow us to rank at least some pairs of sets, i.e., it will allow us to predict the choices of our voter in at least some cases.

### 2.1 The Kelly Principle

According to Kelly (1977), we should prefer a singleton consisting only of  $a$  to a singleton consisting only of  $b$  if we prefer  $a$  to  $b$  (extension axiom); and we should like a set no more than its best element and no less than its worst element. The *Kelly Principle* is thus defined by three axioms:

- (EXT)  $\{a\} \succ \{b\}$  if  $a > b$
- (MAX)  $\{\max(A)\} \succsim A$  where  $\max(A) := a^*$  such that  $a^* > a$  for all  $a \in A \setminus \{a^*\}$
- (MIN)  $A \succsim \{\min(A)\}$  where  $\min(A) := a^*$  such that  $a > a^*$  for all  $a \in A \setminus \{a^*\}$

We write (KEL) for the conjunction of (EXT), (MAX), and (MIN). The Kelly Principle amounts to very weak assumptions: even if we do not know anything at all about the tie-breaking rule or our voter’s attitude towards risk, we can say for sure that she will conform to it.

### 2.2 The Gärdenfors Principle

The *Gärdenfors Principle* states that you should prefer set  $A$  over set  $B$  if you can obtain  $B$  from  $A$  by means of a sequence of operations that involve either removing the most preferred element of the set or adding a new element that is less preferred than those already in the set (Gärdenfors, 1976):

- (GF1)  $A \cup \{b\} \succ A$  if  $b > a$  for all  $a \in A$
- (GF2)  $A \succ A \cup \{b\}$  if  $a > b$  for all  $a \in A$

We write (GAR) for the conjunction of (GF1) and (GF2). Note that whenever  $A \succsim B$  under the Kelly Principle, then  $A \succsim B$  also under the Gärdenfors Principle, i.e., (GAR) entails (KEL). Accepting (GAR) is known to be equivalent to assuming that our voter believes that any ties will be broken by an outside arbiter according to some unknown but fixed preference order (Erdamar and Sanver, 2009).

### 2.3 The Sen-Puppe Principle

Expanding on a proposal by Sen (1991), Puppe (1995) defined the following axiom:

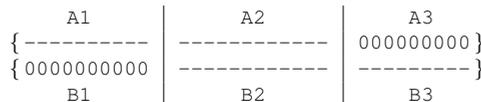
- (SIF)  $(A \setminus \{a\}) \cup \{b\} \succsim A$  if  $b > a$ , with  $a \in A$  and  $b \notin A$

We call this the *single-flip axiom*. It states that replacing an element in a set with another element that is strictly more preferred should result in a set that is at least as good as the original one. Let us call the conjunction of (GAR) and (SIF) the Sen-Puppe Principle. It is more restrictive and less standard than those of Kelly and Gärdenfors, but still weaker than other common assumptions in the literature (Endriss, 2012).

### 2.4 Algorithms for Checking Dominance

Given two sets  $A, B \in \mathcal{X}$  and a total order  $\geq$  on  $\mathcal{X}$ , we now want to devise an algorithm to check whether a given principle of preference extension suffices to infer that  $A$  *weakly dominates*  $B$  ( $A \succsim B$ ) and whether it suffices to infer that  $A$  *strictly dominates*  $B$  ( $A \succ B$ ).

We can represent  $A$  and  $B$  as sequences of 0’s and 1’s. For instance, if there are 5 candidates, then  $\{11001\}$  is the set consisting of our voter’s two most preferred candidates as well as her least preferred candidate. We first discuss the case of strict dominance. For each of the three principles introduced, a necessary precondition for inferring  $A \succ B$  is that we can divide each of the two lists into three (possibly empty) sublists, such that the first two sublists are of equal length, the second two sublists are of equal length (and thus also the third two sublists are), and that the third sublist corresponding to  $A$  and the first sublist corresponding to  $B$  only consist of 0’s. The situation is summarised by the following diagram:



Furthermore, at least one of the lists  $A_1$  and  $B_3$  need to contain a 1. These are necessary requirements under all three principles. In addition, we need to check whether a division into sublists can be found that also satisfies the following principle-specific conditions regarding the two *middle parts* of the list ( $A_2$  and  $B_2$ ):

- (1) We can infer  $A \succ B$  according to the Kelly Principle if and only if both  $A_2$  and  $B_2$  are the empty list for some division into sublists.
- (2) We can infer  $A \succ B$  according to the Gärdenfors Principle if and only if  $A_2$  and  $B_2$  are equal for some division into sublists.
- (3) We can infer  $A \succ B$  according to the Sen-Puppe Principle if and only if the following procedure succeeds for some division into sublists: Move simultaneously through both  $A_2$  and  $B_2$ , from left to right. At each position, apply the appropriate rule:
  - If the remaining two lists to the right of the current position are equal, stop and succeed (that is, succeed immediately if  $A_2$  and  $B_2$  were equal to begin with).
  - If either both lists have a 1 or both have a 0 at the current position, move on to the next position.
  - If there is a 1 at the current position in  $A_2$  and a 0 in  $B_2$ , then flip the next available 1 in  $B_2$  to a 0 (stop and fail if there is no such 1), and move on to the next position.
  - If there is a 0 at the current position in  $A_2$  and a 1 in  $B_2$ , then stop and fail.

The algorithm for checking weak dominance is very similar. There are only two differences. First, it is not necessary for one of  $A_1$  and  $B_3$  to contain a 1. Second, for the Kelly Principle we can also infer  $A \succcurlyeq B$  if for some division into sublists both  $A_2$  and  $B_2$  are equal to a list consisting of just a single 1.

For the sake of brevity, we omit a detailed proof of correctness of these algorithms. For the Kelly Principle, correctness is immediate. For the Gärdenfors Principle, note how the condition on  $A_1/B_1$  corresponds to **(GF1)**, that on  $A_3/B_3$  to **(GF2)**, and that on  $A_2/B_2$  to the fact that  $\succcurlyeq$  is reflexive. For the Sen-Puppe Principle, observe that the rules for the comparison of  $A_2$  and  $B_2$  closely model **(SIF)**.

### 3 Sincerity and Manipulation under Approval Voting

We are now ready to present our results on insincere manipulation under AV.

#### 3.1 The Case of Three Candidates

Let us first analyse the case of three candidates. Suppose a would-be manipulator knows how all other voters are going to vote. Given those votes, one or more candidates will have received the highest number of approvals; we call them the *pivotal* candidates. Some will have received exactly 1 point less than the pivotal candidates; we call them the *subpivotal* candidates. We call the remaining candidates *insignificant*; they have no chance of being elected, however our manipulator is going to vote.

There are  $3^3 - 2^3 = 19$  different *situations*: each candidate must belong to one of the three groups, and at least one of them must be pivotal. Each of the rows in Table 1 corresponds to one such situation. Row (12) with label  $S.P.I.$ , for example, represents the situation where our manipulator's favourite candidate is subpivotal, her second choice is pivotal, and her last choice is insignificant. The columns of Table 1 correspond to the valid ballots available to her.<sup>1</sup> The two sincere ballots are shown on the left; the four insincere ballots are shown on the right. The table cells correspond to the election outcomes for a given situation and a given final ballot. For example, if our manipulator chooses the sincere ballot  $[110]$  (i.e., if she approves of her two most preferred candidates) in situation  $S.P.I.$ , then outcome  $\{010\}$  will be realised (i.e., her second favourite candidate will be the sole winner).

The manipulator knows in which situation we are and needs to choose a ballot. She does so by choosing (one of) the best outcome(s) attainable to her in the situation at hand. Depending on our assumptions on set preferences, we can exclude some possible choices. Suppose all we know is that the manipulator satisfies the Kelly Principle. In Table 1, we have underlined all those outcomes that, according to the Kelly Principle, are not strictly dominated by any other outcome in the same row. This shows that in most situations our voter will have no incentive to vote insincerely. For example, in situation (12) the unique top outcome is

<sup>1</sup>We do not list the *abstention ballots*  $[000]$  (approving no candidates) and  $[111]$  (approving all candidates). By a general result, such abstention ballots cannot influence the type of result we seek here (Endriss, 2012, Lemma 3).

Situation	Sincere		Insincere Ballots			
	[100]	[110]	[001]	[010]	[011]	[101]
(1) P.P.P	{ <u>100</u> }	{110}	{001}	{010}	{011}	{101}
(2) P.P.S	{ <u>100</u> }	{110}	{111}	{010}	{010}	{ <u>100</u> }
(3) P.P.I	{ <u>100</u> }	{110}	{110}	{010}	{010}	{ <u>100</u> }
(4) P.S.P	{ <u>100</u> }	{ <u>100</u> }	{001}	{111}	{001}	{101}
(5) P.S.S	{ <u>100</u> }	{ <u>100</u> }	{101}	{110}	{111}	{ <u>100</u> }
(6) P.S.I	{ <u>100</u> }	{ <u>100</u> }	{ <u>100</u> }	{110}	{110}	{ <u>100</u> }
(7) P.I.P	{ <u>100</u> }	{ <u>100</u> }	{001}	{101}	{001}	{101}
(8) P.I.S	{ <u>100</u> }	{ <u>100</u> }	{101}	{ <u>100</u> }	{101}	{ <u>100</u> }
(9) P.I.I	{ <u>100</u> }	{ <u>100</u> }	{ <u>100</u> }	{ <u>100</u> }	{ <u>100</u> }	{ <u>100</u> }
(10) S.P.P	{ <u>111</u> }	{010}	{001}	{010}	{011}	{001}
(11) S.P.S	{ <u>110</u> }	{010}	{011}	{010}	{010}	{ <u>111</u> }
(12) S.P.I	{ <u>110</u> }	{010}	{010}	{010}	{010}	{ <u>110</u> }
(13) S.S.P	{ <u>101</u> }	{ <u>111</u> }	{001}	{ <u>011</u> }	{001}	{001}
(14) S.I.P	{ <u>101</u> }	{ <u>101</u> }	{001}	{001}	{001}	{001}
(15) I.P.P	{011}	{010}	{001}	{ <u>010</u> }	{011}	{001}
(16) I.P.S	{010}	{010}	{011}	{ <u>010</u> }	{010}	{011}
(17) I.P.I	{010}	{010}	{010}	{ <u>010</u> }	{010}	{010}
(18) I.S.P	{001}	{011}	{001}	{ <u>011</u> }	{001}	{001}
(19) I.I.P	{ <u>001</u> }	{ <u>001</u> }	{001}	{ <u>001</u> }	{ <u>001</u> }	{ <u>001</u> }

Table 1: Outcomes for each possible situation and ballot, for elections with three candidates.

{110} and that outcome is attainable by voting sincerely using [110] (besides being also attainable via one of the insincere ballots). In fact, rather surprisingly, there are *only two critical situations* where this is not the case. These are situations (11) and (13). In situation (11) it is conceivable that our voter’s most preferred outcome is {111} (and not {110}), in which case she would have an incentive to vote insincerely using the ballot [101]. In situation (13), it is conceivable that her most preferred outcome is {011} (and neither {101} nor {111}), in which case she might vote using [010].

The following two “*plain axioms*” are directly derived from rows (11) and (13). They represent the minimal additional assumptions we need to make, above and beyond the Kelly Principle, if we want to rule out any incentives for our voter to vote insincerely.

- (AX1)  $\{a, b\} \succcurlyeq \{a, b, c\}$  if  $a > b$  and  $b > c$
- (AX2)  $\{a, c\} \succcurlyeq \{b, c\}$  or  $\{a, b, c\} \succcurlyeq \{b, c\}$  if  $a > b$  and  $b > c$

For example, (AX2) excludes the possibility that {011} (corresponding to {b, c} in the statement of the axiom) is strictly preferred to both {101} and {111}. Hence, in situation (13), one of the two sincere outcomes that are undominated according to the Kelly Principle will be most preferred amongst all feasible outcomes. To summarise our observations, inspection of Table 1 allows us to establish the following result:

**Theorem 1.** *Under AV with three candidates, upon learning the ballots of the other voters, a voter whose preferences satisfy (KEL), (AX1), and (AX2) will always have a best response that is sincere.*

That is, under the stated (very weak) assumptions on set preferences, no voter will ever have an incentive to vote by means of an insincere ballot when there are (at most) three candidates up for election. We get the same positive result for any other axiom system that entails the axioms referred to in Theorem 1 (for the special case of three candidates). In particular, we obtain the following result:

**Theorem 2.** *Under AV with three candidates, upon learning the ballots of the other voters, a voter whose preferences satisfy (GAR) will always have a best response that is sincere.*

*Proof.* This is an immediate corollary of Theorem 1, because (GAR) entails (KEL), (AX1) and (AX2). □

Theorem 2 is a known result. It immediately follows from Theorem 3 of Brams and Fishburn (1978), whose assumptions on preference extension are equivalent to (GAR). Given that (GAR) is *not* entailed by (KEL) together with (AX1) and (AX2), Theorem 1 is technically slightly stronger than Theorem 2.

- (1) Compute the set  $\Omega$  of *undominated sincere outcomes* for  $S$  as follows:
- (a) For each sincere ballot  $B$ , compute the outcome for  $S$  and  $B$ . Collect all thus computed outcomes in  $\Omega$  (dropping any duplicates).
  - (b) Remove any element  $A$  from  $\Omega$  for which there is another element  $A'$  in  $\Omega$  such that  $A'$  is known to strictly dominate  $A$ .
- Observe that this ensures that the best possible outcomes available to our voter by means of a sincere ballot will be elements of  $\Omega$ , and  $\Omega$  is the smallest set with this property that we can construct.
- (2) For every insincere ballot  $B$ :
- (a) Compute the (insincere) outcome  $A_{[S,B]}$  for  $S$  and  $B$ .
  - (b) If there exists a (sincere) outcome  $A$  in  $\Omega$  that is known to weakly dominate  $A_{[S,B]}$  (note that this includes the case where  $A_{[S,B]} = A$ ), then proceed without producing any output. Otherwise:
    - ▷ Compute the set  $\Omega' \subseteq \Omega$  as the set of all (sincere) outcomes  $A \in \Omega$  such that  $A_{[S,B]}$  is *not* known to strictly dominate  $A$ . (Note that  $\Omega'$  could be the empty set.)
    - ▷ Output the following plain axiom:  $A \succcurlyeq A_{[S,B]}$  for at least one  $A \in \Omega'$ . (Observe that if  $\Omega'$  is the empty set, then this axiom *cannot* be satisfied.)

Table 2: Algorithm to compute additional axioms required to rule out insincere voting for situation  $S$ .

### 3.2 Automated Derivation of Sincerity Results for a Fixed Number of Candidates

When there are four candidates and voters conform to the Gärdenfors Principle, then insincere voting *can* occur. To see this, consider the following example. Suppose your preferences are  $a > b > c > d$ ; candidates  $b$  and  $d$  have received 10 approvals each (they are pivotal); and  $a$  and  $c$  have received 9 approvals each (they are subpivotal). Then you can force outcome  $\{b\}$  (by voting for  $a$ ,  $b$  and  $c$ ), outcome  $\{a, b, c, d\}$  (by voting for  $a$  and  $c$ ), outcome  $\{a, b, d\}$  (by voting for  $a$ ), and several other outcomes that are dominated by one of the first three under the Gärdenfors Principle (e.g., voting only for  $c$  gives  $\{b, c, d\}$ , which is dominated by  $\{b\}$ ). The Gärdenfors Principle is not strong enough to tell us which of those three outcomes you prefer the most. If it is, say,  $\{b\}$ , then we are fine, because you can achieve it by voting sincerely. But it *might* be the case that  $\{a, b, c, d\}$  is your most preferred feasible outcome, in which case you have an incentive to vote insincerely (for  $a$  and  $c$ , but not for  $b$ ).

Hence, we will not be able to generalise Theorem 2 to the case of four candidates. Instead, in analogy to our analysis of elections with three candidates, we now want to analyse *how many* scenarios there are where the Gärdenfors Principle cannot rule out manipulation with an insincere ballot. The problem, however, is that when there are more than three candidates it is not feasible anymore to write out and reliably check the kind of data shown in Table 1 by hand. For four candidates, there are already  $3^4 - 2^4 = 65$  rows and  $2^4 - 2 = 14$  columns to such a table; for five candidates there are  $3^5 - 2^5 = 211$  rows and  $2^5 - 2 = 30$  columns. Instead, we propose to generate and check this kind of data automatically.

Suppose we have fixed the number of candidates  $n$  and the set of assumptions we want to make regarding the extension of preferences to set preferences. Given  $n$ , we can generate all possible *situations* we need to consider: these can be represented by all lists of length  $n$  of the letters  $\mathbb{P}$ ,  $\mathbb{S}$ , and  $\mathbb{I}$  that include at least one copy of  $\mathbb{P}$  each. We can also generate all possible *proper ballots*: these are all lists of length  $n$  of the numbers 0 and 1 that include at least one 0 and one 1 each. Those ballots that are lists in which all occurrences of 0 occur to the right of all occurrences of 1 are *sincere*; all others are *insincere*. Given any situation and any ballot, it is easy to compute the resulting *outcome* (the set of winning candidates). As we have seen in Section 2.4, for any two outcomes  $A$  and  $A'$ , we can check whether  $A$  is *known to weakly dominate*  $A'$  according to the assumptions we have made regarding  $\succcurlyeq$ ; and we can also check whether  $A$  is *known to strictly dominate*  $A'$  under these assumptions.

Now, to systematically check whether a particular scenario admits a situation where a voter would have an incentive to vote insincerely and to generate all additional plain axioms that would be required to rule out any such scenario, we execute the algorithm described in Table 2, once for every possible situation  $S$ . This algorithm will return a (possibly empty) list  $\Gamma$  of plain axioms (from which we can remove any duplicates):

- (1) If the list  $\Gamma$  is empty, then this proves that our assumptions are sufficiently strong to guarantee the absence of incentives to vote insincerely.
- (2) If  $\Gamma$  includes an unsatisfiable axiom (that is, an axiom with  $\Omega' = \emptyset$ ), then this proves that our assump-

tions do allow for situations where a voter will have an incentive to vote insincerely. Furthermore, in this case it is *impossible* to rectify this problem by adding additional axioms.

- (3) Otherwise, voters will sometimes have incentives to vote insincerely, but this problem *can* be rectified. We then have a proof that the original assumptions together with the axioms in  $\Gamma$  will guarantee that no voter ever has an incentive to vote insincerely.

We have implemented the algorithm and the algorithm for deciding dominance under the Kelly, the Gärdenfors, and the Sen-Puppe Principles described in Section 2.4 in PROLOG.<sup>2</sup>

### 3.3 The Case of Four Candidates

In the case of four candidates with voters conforming to the Gärdenfors Principle, our program produces the following two plain axioms as output:

$$\begin{aligned} \text{(AX3)} \quad & \{b\} \succ \{a, b, c, d\} \text{ or } \{a, b, d\} \succ \{a, b, c, d\} && \text{if } a > b > c > d \\ \text{(AX4)} \quad & \{a, d\} \succ \{a, c, d\} \text{ or } \{a, b, d\} \succ \{a, c, d\} \text{ or } \{a, b, c, d\} \succ \{a, c, d\} && \text{if } a > b > c > d \end{aligned}$$

The situation corresponding to (AX3) is S . P . S . P. This is the case familiar from the example of Section 3.2. The situation corresponding to (AX4) is S . S . S . P. In this case, the insincere ballot [1010] will produce the outcome {1011}, which the Gärdenfors Principle alone is not strong enough to show to be at least as preferred as the three outcomes attainable by means of sincere ballots.

Provided our program is a *correct implementation of the algorithm* (on this point, see Section 4), we can infer the following result:

**Theorem 3.** *Under AV with four candidates, upon learning the ballots of the other voters, a voter whose preferences satisfy (GAR), (AX3), and (AX4) will always have a best response that is sincere.*

That is, a minor refinement of the Gärdenfors Principle will rule out any incentives to vote insincerely for a voter who knows how the others are going to vote, even for elections with four candidates. Under the Sen-Puppe Principle the refinement required is even smaller:

**Theorem 4.** *Under AV with four candidates, upon learning the ballots of the other voters, a voter whose preferences satisfy (GAR), (SIF), and (AX3) will always have a best response that is sincere.*

*Proof.* The second disjunct of (AX4) is an instance of (SIF), i.e., the latter entails the former. The claim then follows from Theorem 3.  $\square$

(SIF), and certainly the much weaker (AX4), as well as (AX3) are all reasonable assumptions that will be justified in many practical cases. The Gärdenfors Principle itself is certainly widely accepted. That means, in practice, we will usually be able to exclude insincere manipulations from voters who have obtained full information on the voting intentions of others for elections with (up to) four candidates.

### 3.4 Quantitative Results

As we have seen, when the number of candidates is small, so are both the number of situations in which we must consider it possible that a voter might vote insincerely and the number of additional plain axioms that we would have to accept to be able to rule out insincere voting. Naturally, as we increase the number of candidates, the number of such “exceptions” will go up as well. To give an impression of *how much* the number of exceptions grows, Table 3 provides an overview of the relevant figures for elections with 2–7 candidates for our three preference extension principles. For each principle and each election size, the corresponding table cell shows two figures. The one on top is the number of critical situations (specifying for each candidate whether he is pivotal, subpivotal, or insignificant) in which a voter may benefit from insincere manipulation, and the one at the bottom is the number of additional plain axioms required. In no instance did our algorithm return an unsatisfiable axiom. That is, in all cases it is possible to amend the assumptions we started out with so as to rule out insincere manipulation, albeit in some cases a very large number of additional plain axioms will be required.

For comparison, Table 3 also shows the overall number of situations for each election size (which is  $3^n - 2^n$ ). As we can see, as  $n$  grows, not only does the absolute number of critical situations increase, but the same is true for the ratio of critical situations over situations in general.

<sup>2</sup>The full program required to generate all the results in this paper consist of around 80 lines of code and is available from the author or may be downloaded from <http://www.illc.uva.nl/~ulle/approval-voting/>.

Number of candidates:	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$
Kelly Principle: (KEL)	0 0	2 2	14 19	64 114	244 553	846 2372
Gärdenfors Principle: (GAR)	0 0	0 0	2 2	18 24	100 173	444 972
Sen-Puppe Principle: (GAR) + (SIF)	0 0	0 0	1 1	10 10	61 63	294 321
Number of situations:	5	19	65	211	665	2059

Table 3: Number of critical situations (top figure) and number of additional plain axioms required to rule out insincere voting (bottom figure), for different axiom systems and numbers of candidates ( $n$ ).

## 4 Conclusion

We have seen several results highlighting conditions under which a voter in an approval election who has obtained full information on the voting intentions of all other voters (the classical manipulation scenario) will never have an incentive to vote insincerely. These results for *weak* principles of preference extension and *small* numbers of candidates complement earlier results for *strong* principles of preference extension and *arbitrary* numbers of candidates (Endriss, 2012). Interestingly, our results show that the kind of manipulation considered can usually be ruled out for elections with up to four candidates, while in the classical setting this is only possible for elections with at most two candidates (Taylor, 2005).

Just as interesting as the results themselves is the method we have used to obtain them, namely by automatically exploring the space of all possible voting situations. All of our results have been derived using a very short logic program. The correctness of such a program, and thus of our results, can be verified in a similar manner as a classical manual proof.

## References

- K. J. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, 2nd edition, 1963.
- S. Barberà, W. Bossert, and P. Pattanaik. Ranking sets of objects. In *Handbook of Utility Theory*, volume 2. Kluwer Academic Publishers, 2004.
- S. J. Brams and P. C. Fishburn. Approval voting. *The American Political Science Review*, 72(3):831–847, 1978.
- F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. In G. Weiss, editor, *Multiagent Systems*. MIT Press, 2012. In press.
- Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. In *Proc. 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-2007)*. Springer-Verlag, 2007.
- V. Conitzer, A. J. Davenport, and J. Kalagnanam. Improved bounds for computing Kemeny rankings. In *Proc. 21st National Conference on Artificial Intelligence (AAAI-2006)*, 2006.
- U. Endriss. Sincerity and manipulation under approval voting. *Theory and Decision*, 2012. In press.
- B. Erdamar and M. R. Sanver. Choosers as extension axioms. *Theory and Decision*, 67(4):375–384, 2009.
- P. Faliszewski and A. D. Procaccia. AI’s war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.
- P. Gärdenfors. Manipulation of social choice functions. *Journal of Economic Theory*, 13(2):217–228, 1976.
- C. Geist and U. Endriss. Automated search for impossibility theorems in social choice theory: Ranking sets of objects. *Journal of Artificial Intelligence Research*, 40:143–174, 2011.
- U. Grandi and U. Endriss. First-order logic formalisation of impossibility theorems in preference aggregation. *Journal of Philosophical Logic*, 2012. In press.
- J. Kelly. Strategy-proofness and social choice functions without single-valuedness. *Econometrica*, 45(2):439–446, 1977.
- J. Lang. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence*, 42(1–3):37–71, 2004.
- T. Nipkow. Social choice theory in HOL: Arrow and Gibbard-Satterthwaite. *Journal of Automated Reasoning*, 43(3): 289–304, 2009.
- C. Puppe. Freedom of choice and rational decisions. *Social Choice and Welfare*, 12(2):137–153, 1995.
- A. K. Sen. Welfare, preference and freedom. *Journal of Econometrics*, 50(1–2):15–29, 1991.
- P. Tang and F. Lin. Computer-aided proofs of Arrow’s and other impossibility theorems. *Artificial Intelligence*, 173(11): 1041–1053, 2009.
- A. D. Taylor. *Social Choice and the Mathematics of Manipulation*. Cambridge University Press, 2005.
- F. Wiedijk. Arrow’s Impossibility Theorem. *Formalized Mathematics*, 15(4):171–174, 2007.

# Greed, Envy, Jealousy

## A Tool for more efficient Resource Management

M.D. Farjam <sup>a</sup>

W.F.G. Haselager <sup>b</sup>

I.G. Sprinkhuizen-Kuyper<sup>b</sup>

<sup>a</sup> *Department of Artificial Intelligence, Radboud University Nijmegen, The Netherlands*

<sup>b</sup> *Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour, The Netherlands*

### Abstract

Highly social animals like humans developed features such as greed, envy, and jealousy through evolution. Assuming that the concept of envy has already been learned, experiments are performed in an artificial life environment. They show the benefits of envy for a multiagent system and how principles underlying envy can make agents more effective with respect to resource management. Furthermore they show under which circumstances (such as the population size or the possibility to punish greed) jealousy turns into a useful feature in a multiagent system. Concepts like population size or availability of resources are translated back into real world phenomena to show possible applications of artificial envy. Simulations show that the benefits in resource-management outweigh the costs of having an envy system.

## 1 Introduction

We find greed, envy and jealousy in highly social animals and across human cultures [6]. The question is why jealousy evolved as a universal feature in social interaction while it is viewed as something negative. We propose that jealousy evolved because it brings some advantage to the species. This point was made explicit for the first time by Wilson [9] and modifications of this point are popular in modern sociobiology and psychology [5].

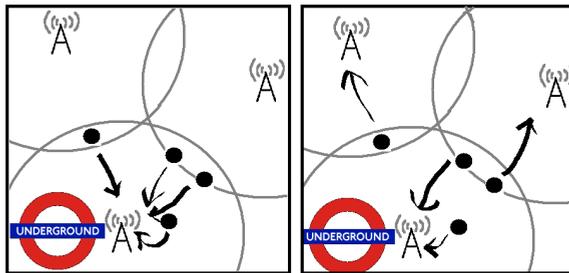
We aim to show through artificial life how greed, envy, and jealousy influence resource management in societies. Observations made in an artificial life environment can be transferred to programs that autonomously share resources such as calculating capacities and make them more efficient. In the following paragraphs we will first show what we mean by jealousy and related terms, then how we simulate jealousy. Finally we will describe how we expect jealousy to influence simulations we want to perform in an artificial life environment.

To ascribe a mental state, such as envy someone, to agents we use two criteria. Not just the behavior must be the same as in humans that we call jealous, but also the reason(s) for the behavior [3]. Humans envy because they think that others have something they deserve more [8]. Humans adjust their behavior (e.g. some kind of punishment) through which they express envy to the specific situation. Behavior depends, e.g. on the costs of punishing or on the general disparity in the environment. These kind of factors will be incorporated into our simulations to understand how envy interacts with such factors. Later on, it will be shown how to translate these factors back, not only to problems in artificial intelligence, but also to sociological and psychological questions. By doing so, we hope to find mechanisms through which resource management (distribution of things valuable to an agent) becomes more effective.

In the experiment conducted we assumed that being jealous already has been learned by agents. This means that agents in the artificial-life environment will automatically punish other agents in their neighborhood when they are richer (above a certain threshold) than themselves. Furthermore agents will try to prevent punishment by sharing with agents that are much worse off (above the same threshold) than themselves. This behavior occurs automatically and stays the same during the simulation. Agents do not learn or adapt behavior during the simulations.

Throughout this paper we will refer to two scenarios/real life situations to which we can apply the concepts, problems, and actions occurring in the simulations. Through these scenarios the reader on the one hand will see how certain concepts in this experiment related to our social reality, and on the other hand see possible applications of the findings. These scenarios are not intended to prove anything. They surely have their shortcomings and may be interpreted differently. However they show the external validity of this paper.

Scenario 1 describes the social reality we encounter daily. Humans live in groups where some possess more than others. Envy and dissatisfaction occurs when disparity is too high. People can punish greedy and wealthy people in many ways such as excluding them socially or trying to take them to court. In both cases punishment will cost both parties resources (e.g. money, time, friends). People in such societies often try to avoid the negative consequences of envy and instead prefer to decrease disparity by sharing some of their resources (through such institutions as foundations, social welfare, amicable agreements). Humans also deviate in the amount of money they need to survive. For instance, while earning amount  $X$  as a single may be sufficient, a father of a large family may need much more to make ends meet.



**Figure 1:** Scenario 2 (dots are phones, arrows show the tower used)

Scenario 2 is taken from the field of mobile communications. Mobile telephones find the broadcasting tower that is closest to them and try to connect to it. At some point some broadcasting towers may be doing nothing whereas others at another place (e.g. a subway station) are chronically overtaxed. It would be more efficient if towers found a way to balance the work in a way that provides a good connection to every mobile phone without overtaking any one connection point. Figure 1 shows a scenario where towers are doing nothing although they could easily “help” (share their free capacities) another tower by taking over some of the other tower’s mobile phones, even though they are not closest to these phones. Making towers envy other towers, which have more free capacity, could help.

De Jong [1] showed in his PhD thesis that fairness is beneficial for the total reward of a group when resources are shared. Humans automatically and often subconsciously punish unfair behavior to force individuals to act fair. Punishment can be social (e.g. excluding individuals from the group) or material (e.g. a penalty or fine). De Jong showed in his experiments that individuals will be more egoistic if there is no punishment. On an inter-agent level we need possibilities to punish selfish behavior to rein in egoistic, destructive behavior within the group.

On an intra-agent level there seems to be an intuitive link between greed and envy. Why envy someone when one does not want more? The next step towards envy is to make our agents “feel” less content with something they have, while observing others who have more. Fehr and Schmidt [2] developed a utility function with exactly these properties. Through their utility function they can explain why we often feel better off getting no reward, than getting a small reward while observing others getting a much higher one.

Humans attribute mental states to themselves and others and are thereby able to predict the behavior of others [7]. By knowing what envy is and the assumption that others “feel” envy for the same reasons, they are able to predict when envy will occur as a result of their actions. If it is general knowledge that envy will lead to punishment against the one that is envied, agents will take the punishment in consideration for their actions and expected utility.

We programmed an artificial-life environment in which agents try to gather (greedily) as much resources as possible. Furthermore they have the possibility to punish agents that have more resources than themselves or share resources when others have less. Agents will die after a certain time, but can prolong their life when they get resources.

The question remains why one should make the effort to implement greed and jealousy and not simply

let agents share resources with each other. There are at least two big advantages to taking this detour to sharing. First of all, sharing resources very often consumes resources. When a computer decides to take over a computation initializing variables will consume computation time. The same holds for humans. Sometimes it is more effective to do something ourselves, than finding someone else who has more time. Adding an agent's egoistic perspective while deciding whether it will benefit from sharing may ultimately spare resources. The second advantage of envy above simple sharing is that envy will lead to punishment and punishment provides a possibility to learn effective sharing. In a training phase, agents would learn through punishing each other how they can avoid being punished through sharing. Even after training, punishment will enable agents to adapt their sharing behavior to a changing environment.

We propose that envy, although it has a bad reputation, will make resource management more effective than simple greed alone without envy. How does resource management change in a multiagent system when there is envy and punishment? We will take a close look at the behavioral changes per agent, and the changes within different kinds of environments or agent populations. Specific attention will be paid to identify the environments and parameters that make jealousy most effective.

## 2 Method

### 2.1 Software

We used the open-source software *breve* 2.7.2 [4] to implement an environment with jealous agents. Simulations start with random values of the parameters. For later statistical analysis, a record is kept of the age each agent in a simulation reached and the parameters with which the simulation was initialized. All parameters (see Table 1) of the simulations will be explained in the following paragraphs.

**Table 1:** Parameters used in the simulation and the corresponding phenomenon in the scenarios.

Concept in simulation	Values	Scenario 1	Scenario 2
jealous	All 'yes' or all 'no'	jealousy	active jealousy algorithm
populationDensity	0.5, 0.83, 1.16, 1.5	humans in area	number of towers in area
fruitsAvailable	0.15, 0.45, 0.75, 1.05	e.g. GDP	1/(phones in area)
actionEffectRatio	1.0, 7.6, 13.6, 20	legal costs	computation needed action
heterogeneityRescNeed	0, 0.66, 1.33, 2	people depending on income	crowded areas – calmer areas
toleranceHeterogeneity	1.0, 7.6, 13.6, 20	feeling about acceptable disparity	accepted difference in workload

### 2.2 The environment

All simulations were conducted in an identical square environment. Here fruits (representing resources) pop up randomly. Fruits start with a certain energy that decreases with every iteration of the simulation. Fruits disappear when the energy is zero. The number of fruits that pop up per iteration depends on the variable *fruitsAvailable* and varies between low (if not eaten by an agent 0.15% of the area has fruit on it), midlow, midhigh and high (1.05% of the area is covered with fruits). Agents will be set in the environment at random positions. The number of agents varies between simulations, with 4 possible values for *populationDensity* (between 0.5% and 1.5% of the area is covered by agents initially).

### 2.3 Agents

Initially agents move randomly and have the same amount of energy. Within a fixed neighborhood, agents observe and interact with the environment. Agents strive towards fruits within the neighborhood and by reaching them they eat them (incorporate the energy that is left in the fruit).

With every iteration of the simulation, agents lose energy. Agents will die as soon as their energy level reaches 0. The average age that agents reached before dying is the performance measure of the simulation. Possible values for the simulation's variables are chosen such that the amount of fruits which can be consumed will not exceed the amount of energy that is lost in the long run. So every agent will die.

On average, agents in every simulation consume the same amount of energy per iteration, but per agent there is a deviation from the average possible, depending on the variable *heterogeneityRescNeed*. Four

values are possible ranging from 0 (every agent within the simulation consumes the same amount of energy per round) to 0.2 (the energy consumed ranges between 80 and 120 % of the average).

Besides eating and moving agents can interact with other agents in their neighborhood. The neighborhood is defined within a radius of  $1/50$  of the edge length of the simulation area. Within their neighborhood agents know the energy level of other agents and can compare it to their own. Agents share resources with the poorest neighbor when the difference is too high and positive and if they will have more than 50 energy points left after sharing. When the difference is too high, negative and agents would have a reserve of less than 50 energy points left after sharing, they punish the agent that is the richest in their neighborhood (expressing jealousy). Whether a difference is seen as too high or not depends on the variable toleranceHeterogeneity. The variable can have four values ranging from 0–20. A value of 0 means that even the smallest difference will lead to action, while a value of 20 means that action will only be taken when the difference exceeds 20. Punishing and sharing cost energy. How expensive an action is depends on the variable actionEffectRatio with 1, 7.6, 13.6 and 20 as possible values. A value of 1 means that punishment will cost the punisher as much as the punished, and a value of 20 means that punishing will only cost  $1/20$  of the effect it has on the punished.

## 2.4 Simulations

The above mentioned 6 independent variables lead to a total of 2048 possible combinations of the variable values. Every simulation received randomly chosen values for the independent variables. In total 248 simulations were performed and together imitated about 10000 agents. For every agent the age reached + the parameter values of the environment it lived in (in terms of the independent variables) were stored for later analysis. In each simulation agents got a random starting position and throughout the simulations fruits were positioned randomly.

## 3 Results

W.r.t. the distribution of age the jealous and not-jealous group had a high skewness meaning that many agents died early while some became very old. This is sensible as there are more resources left for one agent as others die. It is remarkable that the jealous group scored higher in skewness (2.8) and kurtosis (9.0) than the not-jealous group (1.5 and 5.2 respectively). This means that age is more homogeneously distributed in the not-jealous simulations than in the jealous ones. Taking a closer look at the histograms (Figure 2, left) it appears that this is the result of a larger (still small) group of agents in the jealous group who live much longer than the average agent. It seems that the agents who live longer (those with low energy costs per iteration) benefit from the remaining agents more when they are jealous and become more effective in their resource management through jealousy in the later stages of the simulation. If we leave out the highest quartile of agents in both groups the distribution and average is almost the same. Although the data is not normally distributed it resembles normality. The size of the sample used for analysis is large and t- and F-tests are robust to violations of a normal distribution when sample size gets large. Therefore they still were performed, but we used 0.01 as our significance value.

A t-test was performed ( $p < 0.001$ ), showing that the group of jealous agents performs better in terms of average life length ( $M = 373.37$ ) than the not-jealous group does ( $M = 351.39$ ). A Mann-Whitney U test shows the same results. The average rank was 4172.54 (jealous) versus 6002.46 (not-jealous) ( $p < 0.01$ ).

We used an ANOVA to analyze the role of each independent variable. The overall tests showed that all variables together explain the variances of age ( $p < 0.001$ ). Every variable had a significant influence on the average age. Furthermore all differences between values of variables were significant (always  $p < 0.001$ ). This is not surprising because of the large sample size. All independent variables and their interactions explained 25% of variance found in the age of agents. Furthermore we only looked at unique effects of variables and all two-way interactions with jealousy as they together explained 17% of the variance of age. This means that an agent's life was influenced greatly by the random factors in the simulations (initial position, position of other agents and location of fruits). Hence effect sizes found for the effects of each variable and the interaction of the variables are likely small. The largest effect sizes of any individual variable was found for jealousy (0.02) and toleranceHeterogeneity (0.021). Effect sizes for actionEffectRatio (0.014) and fruitsAvailable (0.018) were also quite high, whereas populationDensity (0.003) and heterogeneityRescNeed (0.004) had a fairly small effect size.

### 3.1 General Findings

For all values of the independent variables (except for the lowest value of `actionEffectRatio` agents had a higher life expectancy in the jealous group compared to the not-jealous group. No matter what the values of the variables were, jealousy between agents had a positive influence on an agent's age.

Another general effect of jealousy appears to be that it flattens the distribution of age (Figure 2, left). For most independent variables and their values there is a peak of agents dying at an age of  $\sim 300$  iterations. This peak is lower for most independent variables in the jealous group compared to the not-jealous group. In the following paragraphs we will zoom in on independent variables individually.

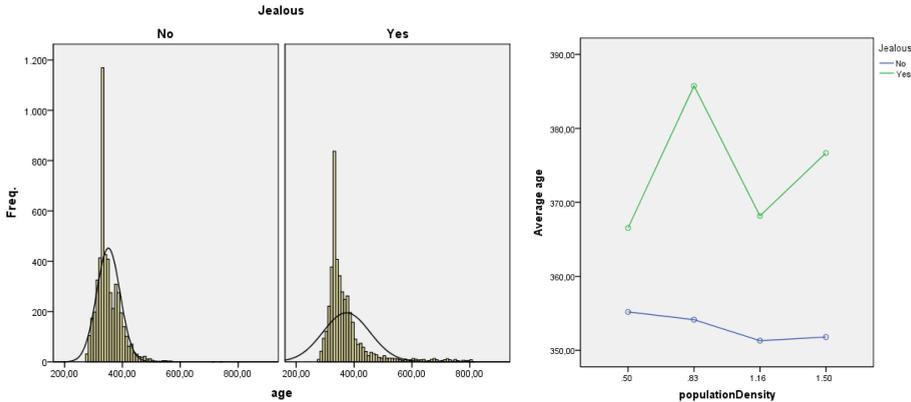


Figure 2: Distribution of age (left); Interaction-plot of age and populationDensity (right).

### 3.2 Population density

In the group of not-jealous agents, average life expectation gradually decreases with an increase of population density. This seems plausible as there is less to eat for every agent when there are more other agents. Agents in the jealous group live longer for all values of population density than the agents in the not-jealous group. Interestingly, the average age of agents in the jealous group increases with a higher population density. However there seems to be no linear relationship between populationDensity and age.

Figure 2 (right) shows a fluctuation in average age from 0.83 to 1.16 in the jealous group. This difference is significant. We can only speculate on the reason for this fluctuation and its implications. We expect that the interaction of jealousy and populationDensity interacts with variables such as `fruitsAvailable` (availability of resources) on these values. However, as described before, we do not look at higher order interactions.

### 3.3 Availability of resources

Both groups profit from a higher availability of resources (`fruitsAvailable`). More resources means more energy per agent and thereby a longer life. Again all values of the variable show a higher age for the jealous group. The difference is stable for almost all values of the variable `fruitsAvailable`.

### 3.4 Tolerance for heterogeneity

As with `actionEffectRatio` the variable `toleranceHeterogeneity` only influences the behavior of agents that are jealous, there the average age stays almost constant at  $\sim 352$  iterations for the not-jealous group.

A tolerance of 0 leads to the highest difference between the jealous and not-jealous group. This difference is the highest (55 iterations) of all differences found in the comparison between jealousy and the other independent variables. It seems that no tolerance for disparity makes jealousy the most effective. The two-way interaction was second in terms of effect size (0.009). When agents become too tolerant for disparity in their neighborhood there is almost no difference between the two groups. The relationship between tolerance and age in the jealous group appears to be linear and negative, suggesting that jealousy becomes

less effective as agents tolerate disparity. Although Figure 3 (left) suggests that jealousy should occur immediately, Figure 3 (right) tells us that jealousy will lead to suboptimal results when punishing gets too cheap. In other words ‘be jealous quickly, but don’t let your actions always be determined by it’.

### 3.5 Action-effect ratio

The not-jealous group stays almost constantly at an average age of 352 (Figure 3, right), which is sensible as this variable does not affect the behavior of not-jealous agents. Here we find the only case where not-jealous agents outperform jealous agents. At a very low ratio the costs of jealousy carries more weight than the benefits of sharing. Furthermore the data suggests that there is a value for the ratio at which it maximizes the advantages of jealousy. Whether this is a local or global maximum cannot be seen from the data. At the ratio of 13.67 we find the second largest difference (43 iterations) between the jealous and not-jealous group, not just for the variable `actionEffectRatio`, but also for all other sub groups that can be made within independent variables. Although small, the `actionEffectRatio` variable had the largest two way interaction with jealousy (effect size = 0.01).

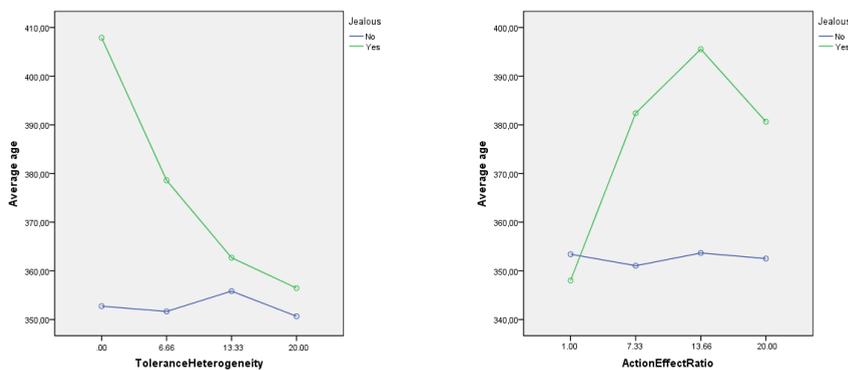


Figure 3: Interaction-plots between age and toleranceHeterogeneity (left) and actionEffectRatio (right)

### 3.6 Heterogeneity resource need

As with all other independent variables the jealous group outperforms the not-jealous group at all values of the variable. The data does not show any bigger interaction between jealousy and heterogeneityRescNeed. Accordingly, this interaction had the smallest effect size of all two-way interactions (0.002). The distributions of the jealous and not-jealous groups show no major differences.

### 3.7 Decision tree

An algorithm was used to build a C4.5 decision tree that decides whether a certain setting of the independent variables will lead to above average lifetime of agents ( $M = 363$ ). The training set consisted of 108 randomly chosen simulations with jealousy. The performance of the decision tree, tested with the remaining 16 jealousy simulations not used before, was 94% (with a depth of 3) for simulations with jealousy. Through the decision tree one can see the parameters that are most important for making jealousy effective. Figure 4 shows the decision tree and its classifications, where a value of 1 stands for the classification that the agents will have an above average age.

The decision tree shows that the most important variable to predict whether a jealous population will become old is the availability of resources ('R' in figure 4). This makes sense as there is an intuitive link between food (resources) and survival (age). More interesting though is going down one level on the tree. On the left side we can see that jealousy can help a population to live longer even when availability of resources is low. By keeping the costs for punishing/sharing low (thus the `actionEffectRatio` (AER) high) 6 out of 11 populations can overcome the negative effects of a low availability of resources through jealousy. Numbers get even better if we consider not just the `actionEffectRatio` but also the `toleranceHeterogeneity` (TH) within the population. When TH is low and AER is high, 6 out of 7 populations will have an above

average life expectation. In the not-jealous group this was only true for 1 out of 16. The right part of the tree does not give much information, except for the fact that heterogeneityRescNeed (HRN) needs to be carefully chosen.

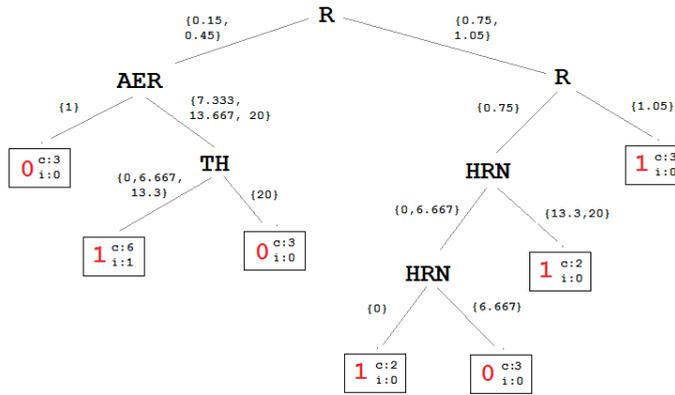


Figure 4: Decision tree for simulations with jealousy (variable values in brackets)

## 4 Discussion

The simulations support the hypothesis that jealousy can help to make autonomous resource management more effective. The simulations also show how jealousy interacts with certain other parameters. Through the two scenarios (see Introduction) we will discuss the meaning of the results.

In Scenario 1 we draw parallels between the simulations and life in human societies. The results suggest that jealousy has a worse reputation than it deserves. It can help to share resources in groups. To make jealousy most effective costs of expressing it (actionEffectRatio in the simulation) should be low. Humans express jealousy by social exclusion of the envied or in extreme cases through legal actions. Lowering the costs of legal actions for poor people is, obviously for a good reason, part of at least some European societies. Another important finding is that the positive effects of jealousy unfold best when the tolerance for disparity is low (toleranceHeterogeneity). This is an important argument in public debates about social redistribution (a society that does not tolerate disparity at all would be communistic). The findings hold for societies with a high diversity in resource needs as well as for those with quite homogeneous resource needs. From a political point of view, the fact that jealousy keeps the average wealth stable with increasing population is also very interesting in our densely populated world.

In Scenario 2 we show how multiple broadcasting towers decide which phones in their area to take over and which to leave for other towers. The amount of calls a tower has to handle is comparable with the resource need of agents in the simulations. Resources are free capacities of towers (negatively correlated to phones in the area). The simple heuristic “every phone gets assigned to the tower that is closest” is not very effective. If towers could be jealous towards other towers that have more free capacity than they do, towers could get a better arrangement (higher connection quality per phone). Similar to the findings in Scenario 1, towers should not be too tolerant to differences in free capacity for jealousy to have maximum effectiveness (low toleranceHeterogeneity). Furthermore, the algorithms that implement the punishment should not require too many resources (high actionEffectRatio), otherwise jealousy may have negative consequences for overall connection quality. The findings for population density suggest that effectiveness of jealousy does not depend on the density of towers in the area (jealousy was effective with any populationDensity).

The question remains whether we can say that jealousy is effective just because the jealous group had a higher average life length. There are cases where we are interested in the average performance of a multi-agent system or a society but there are also scenarios where it is the median that is most important. In Scenario 2 extreme good quality for some phones, but bad quality for many may be worse than a mediocre quality for most phones. One could argue that the effect sizes found are very small (especially for experiments in such a controlled environment) and that the results therefore are not relevant. However, despite

the small effect sizes, jealousy boosted performance of agents by 20 iterations (7%) in all simulations and under optimal circumstances by even more than 50 iterations (20 %). An additional reason for the small effect sizes might be that, while programming the environment, the simulations included as many random variables (location of fruits, location of agents, etc.) as possible to make them more realistic as a model, while still being applicable for dynamic environments. In the two scenarios described there may be many comparable random factors.

In our analysis we only looked at unique effects of the independent variables and two-way interactions with jealousy. Future research will help us to get a more realistic idea of the effects of jealousy by looking at higher-order interactions with jealousy and other independent variables.

A general remark we have to make on the data used and the conclusions drawn is that the dataset did not contain all possible values for the independent variables. The simulations used just a limited range of values for each variable. Relations that seem linear may just be a part of the function that describes the relation between the independent variable and age. Furthermore, no conclusive statements about optimal values for these variables can be made, as maxima that were found may just be local. Limiting the range of variables (like fruitsAvailable) was needed though to make sure that agents die eventually.

Lastly, the simulations assumed that being jealous was already learned. The programmed agents could not adapt their behavior. It was not needed as the environment in the simulations was static. But the two scenarios described have an environment that changes permanently: population density changes, amount of work changes, etc. The question how one can envy flexibly will make for further interesting study.

Many other independent variables and higher-order interactions have to be analyzed to get a deeper understanding of jealousy. The amount of variables and interactions analyzed here was quite small as this paper was meant to make a first attempt towards realizing resource management through jealousy. Critics can also argue that the operationalization and implementation of jealousy has its shortcomings (which it surely has), but nevertheless this paper showed that the implementation of a rough concept of jealousy can increase performance. Not just that, jealousy allows agents to weigh their own interests against the interests of others. We have to keep in mind that all computations were done locally and autonomously by the agents, computations have to be fairly easy for punishment to provide a learning mechanism through which an agent can learn to be jealous in a changing environment. This all makes jealousy a concept that can effectively increase performance in multi-agent systems and we encourage others to study the effect of jealousy on resource management further . . . we promise not to be jealous!

## References

- [1] S. De Jong. Fairness in multiagentsystems, 2009. Unpublished doctoral dissertation, University Maastricht, The Netherlands.
- [2] E. Fehr and K. Schmidt. A theory of fairness, competition and cooperation. *Quarterly Journal of Economics*, 114:817–868, 1999.
- [3] L. Fraley. Strategic interdisciplinary relations between a natural science community and a psychology community. *The Behavior Analyst Today*, 2:209–324, 2001.
- [4] J. Klein. BREVE: a 3D environment for the simulation of decentralized systems and artificial life. In *Proceedings of the eighth international conference on Artificial life*, ICAL 2003, pages 329–334, Cambridge, MA, USA, 2003. MIT Press.
- [5] F. Lelord and C. André. *La Force des émotions*. Poche, Paris, 2003.
- [6] P. Morris, C. Doe, and E. Godsell. Secondary emotions in non-primate species? behavioral reports and subjective claims by animal owners. *Cognition and Emotion*, 22:3–20, 2008.
- [7] D. G. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1:515–526, 1978.
- [8] B. Russell. *The Conquest of Happiness*. H. Liverwright, New York, 1930.
- [9] E. Wilson. *Sociobiology: The new Synthesis*. Belknap Press of Harvard University Press, Oxford, England, 1975.

# *Preserving Precision* as a Guideline for Interface Design for Mathematical Models

Linda C. van der Gaag<sup>a</sup>    Hermi J.M. Schijf<sup>a</sup>  
Armin R. Elbers<sup>b</sup>    Willie L. Loeffen<sup>b</sup>

<sup>a</sup> Department of Information and Computing Sciences, Faculty of Sciences,  
Utrecht University, Utrecht, The Netherlands

<sup>b</sup> Central Veterinary Institute of Wageningen UR, Lelystad, The Netherlands

## Abstract

In collaboration with two veterinary experts, we are in the process of developing a Bayesian network for the early detection of classical swine fever in pig herds. The network is intended for use by veterinary practitioners upon visiting a pig farm with clinical problems of unknown origin. For tailoring the data-entry interface of the network to these prospective users, we embraced well-known design guidelines, which recommend for example employing the users' professional language and hiding the network's technical details. While highly valuable in themselves, we found that these design guidelines did not suffice for guaranteeing that entered information conforms to the mathematically precise meanings of the stochastic variables of our network. In this paper, we supplement best practices for interface design with a new guideline called *preserving precision*, which emerged from our experiences with designing the data-entry interface for the non-mathematical users of our Bayesian network for classical swine fever.

## 1 Introduction

In collaboration with two experts in veterinary science, we are developing a Bayesian network for the early detection of classical swine fever (CSF) in pig herds. The network is aimed at use by veterinary practitioners upon visiting a pig farm with clinical problems of unknown origin. The intended use on a farm site had a number of implications for the data-entry interface to our network. Most importantly, the hardware to be used had to be shockproof, easily portable, and to allow disinfection. Because of these demands, we decided to use a personal digital assistant (pda) and to enclose it in a padded air-tight box in a disposable Ziploc<sup>TM</sup> bag which still allowed interaction with its touch screen. Because of the small screen involved, this decision posed a further challenge for the design of the data-entry interface to our Bayesian network.

For designing user interfaces to automated systems in general, numerous principles and guidelines have been developed, with which considerable experience has been gained for a large range of software products [1, 2]. Since an (implemented) Bayesian network in essence is a software product, we embraced these well-known principles and guidelines for the design of the data-entry interface for our network in classical swine fever. Most prominently, we built on the guideline promoting compatibility with the professional language of prospective users. When developing the interface, we experienced however, that none of the available guidelines provided for explicitly ensuring that the clinical data that would be entered by a veterinary practitioner conformed to the mathematical precise meanings of the stochastic variables of our Bayesian network. Since the high-precision probabilities computed from the network assume these exact meanings, any misinterpretation or imprecision in the entered data could ultimately lead to erroneous conclusions. In each phase of the design of our data-entry interface we thus had to ensure that the mathematically precise meanings of the stochastic variables involved were conveyed to the veterinary users as well as possible, leaving little room for misinterpretation or imprecision. From our experiences thus emerged a new guideline for interface design which we coined *preserving precision*. This new guideline is tailored to the design of interfaces for non-mathematical users to automated systems which build upon mathematically precise concepts and for which mathematical precision needs to be retained in any user interaction.

In this paper we describe the design of the data-entry interface to our Bayesian network for the early detection of classical swine fever in pigs, and thereby present a case study in the design of non-mathematical user interfaces for automated systems including mathematically precise models. We detail how well-known guidelines for interface design in general were employed, and formulate our new guideline of *preserving precision*. We show how we involved prospective users of our Bayesian network in the design of the interface, and in ensuring, more specifically, that the precise meanings of the network's stochastic variables are retained in all user interactions. We further report some preliminary results from an evaluation study of the ability of the designed interface to preserve mathematical precision throughout a field trial of every-day use.

The paper is organised as follows. In Section 2, we provide some background of our Bayesian network for the early detection of classical swine fever. In Section 3, we review well-known guidelines for user-interface design in general and formulate our new guideline of *preserving precision*. In Section 4 we describe how we aligned the contents of the data-entry interface to our network, with the professional language and workflow of its prospective users. Section 5 reports some results from a preliminary evaluation of the interface. The paper ends with our conclusions and directions for further research in Section 6.

## 2 Background of the Classical Swine Fever Project

In the context of a European project, we are developing a Bayesian network for the early detection of classical swine fever (CSF) in pigs. Classical swine fever is a highly infectious viral disease, with a potential of rapid spread. When a pig is first infected with the CSF virus, it will show an increased body temperature and a sense of malaise, associated with such clinical signs as a lack of appetite and lethargy. Later in the infection, the animal is likely to develop an inflammation of the intestinal tract; also problems with the respiratory tract are beginning to reveal themselves through such signs as a conjunctivitis, snivelling, and coughing. The final stages of the disease are associated with problems of the circulatory tract, giving rise to cyanosis and pin-point bleedings, and with lesions of the central nervous system. The accumulating failure of body systems will ultimately cause the pig to die. The longer a CSF infection remains undetected, the longer the virus can circulate without hindrance, not just within a herd but also between herds. Since classical swine fever is a notifiable disease for which eradication measures are installed, an outbreak will have major socio-economic consequences, not just for the farmer involved but also for the nation as a whole.

Although early detection of classical swine fever is of vital importance, the aspecificity of especially its early signs causes a clinical diagnosis to be highly uncertain for a relatively long period of time. The aim of the CSF project now is to supply veterinary practitioners with an additional tool for identifying suspect patterns of disease as early on in an outbreak as possible. For this purpose, a decision-support system is being developed which includes a Bayesian network for reasoning about the uncertainties involved in the clinical diagnosis of classical swine fever. The graphical structure of the Bayesian network for the identification of CSF in individual animals, is depicted in Figure 1; it includes 32 stochastic variables with two to five values each, for which over 1100 (conditional) probabilities are specified.

The decision-support system is intended for use by veterinary practitioners when visiting a pig farm with clinical problems of unknown origin. For consulting the system, a veterinarian has to provide evidence on the health status of a pig, by entering some 15 clinical findings for the stochastic variables in the embedded Bayesian network; for each of these variables, the veterinarian basically has to choose one from among a pre-defined set of possible values. Given the entered evidence, the network computes the probability of a CSF infection being present in the examined pig. Based upon this computed probability, the decision-support system will provide a recommendation for further actions to be taken.

## 3 Guidelines for Interface Design and Their Application

For designing user interfaces in general, numerous guidelines have been formulated, with which considerable experience has been gained for a wide range of software products [1, 2]. For the design of the data-entry interface to our Bayesian network for the early detection of classical swine fever we could therefore build upon a wealth of best practices by embracing these guidelines. We briefly review the four guidelines which proved especially valuable in our context, and describe how we applied them.

The first of the four design guidelines of special interest states that any information presented in the interface should align with the mental model of its prospective users, that is: *know thy user* [2]. Well-known heuristics bearing on this guideline are *speak the user's language* [1] and *workflow compatibility*

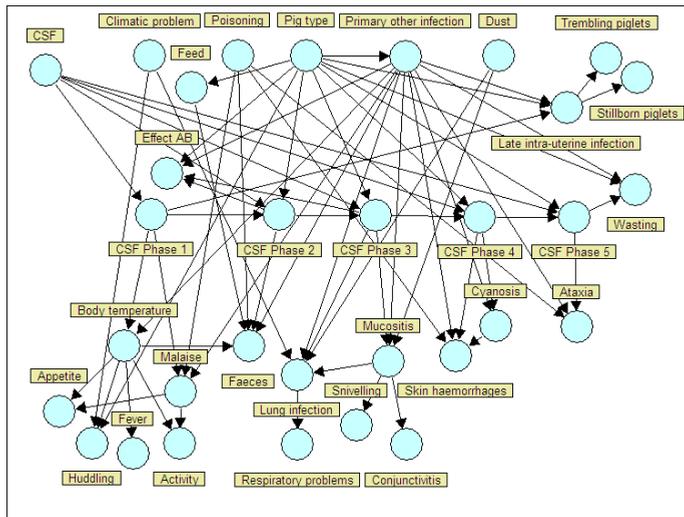


Figure 1: The graphical structure of the Bayesian network for the early detection of CSF in individual pigs

[2]. These heuristics amount to the recommendation to use the professional language of the intended users in any interaction, rather than the language coined by the interface developers. They further recommend to carefully align the flow of information in the interface to the workflow of the projected users in their professional settings. In view of designing the data-entry interface to our Bayesian network, this first guideline basically recommends that the interface should not mention stochastic variables and values, but should present information to its veterinary users in terms of pigs and findings from clinical examinations. The closely related second guideline amounts to the recommendation to use simple and natural dialogue which is unambiguously understood by the prospective users [1].

The third guideline that proved particularly useful for our veterinary context, states that effective interfaces should not concern their users with the inner workings of the system [3], thus promoting *invisible technology* [2]. In view of our Bayesian network, this guideline implies that the veterinary practitioners who have to provide the network with clinical information do not need to be confronted with probability theory or with the graphical representation of the network during data entry. Interestingly, most Bayesian-network tools to date offer a graphical interface which depicts the graphical structure of a network, composed of circles and arrows; often moreover, the (prior or posterior) probabilities computed from the network are represented as bar graphs with the variables. We note that this commonly-used interface clearly does not comply with the invisible-technology guideline. This interface was designed originally by computer scientists, for the purpose of studying the details of a network and its performance. While highly appropriate for network developers, an interface composed of circles, arrows and bar graphs hardly constitutes a suitable interface for veterinary practitioners for example, without any mathematical or formal computer-science background. Their goals are not to study the details of the network, but to arrive at a correct diagnosis by entering clinical data and understanding the network's output. Despite the ease of interpretation which is often acclaimed by network developers, experiences show in fact that even users who had helped develop a Bayesian network have difficulty interpreting the graphical network representation of their knowledge [4].

The last guideline of special interest for our context, is to make the application as efficient as possible. Well-known design heuristics bearing on this guideline are *look at the user's productivity, not the computer's* [3] and *ease of learning and ease of use* [2]. This efficiency guideline proved particularly useful for the data-entry interface to our network since veterinary practitioners are demanding users whose time is precious and who are not well acquainted with the use of automated support in pig barns on site. The faster and more conveniently they can enter any requested clinical data into the decision-support system, the better they will comply with its demands and the less inclined they will a priori be not to use the system.

While a Bayesian network is a software product just like any other type of computerised system, it differs from most software products in its precise mathematical meaning. An interface to such a network has to ensure that this precise meaning is retained throughout all interactions with its users. Any imprecision in entered data could in fact lead to erroneous conclusions being drawn from the high-precision probabilities

which are computed from the network under the assumption of its mathematical definitions. For our network for classical swine fever for example, a user has to be aware of the precise definitions of the stochastic variables involved upon entering clinical findings. To the best of our knowledge, no guideline has been formulated for interface design as yet, to guarantee that mathematical precision is retained. We therefore propose a new guideline to this end, which we coined *preserving precision*. The new guideline pertains especially to the design of interfaces for non-mathematical users to automated systems which build upon mathematically precise concepts and for which mathematical precision needs to be retained in any user interaction. In view of a Bayesian network for a real-world problem domain for example, the guideline states that an interface should guarantee that its users interpret all exchanged information in the sense of the precise definitions of the underlying variables and values.

## 4 Developing the Data-entry Interface for the CSF Network

Our Bayesian network for the early detection of classical swine fever is aimed at use by veterinary practitioners when visiting a pig farm. To meet the demands of on-site use, the data-entry interface to our network was to be programmed on a personal digital assistant. We now describe our considerations upon developing the interface, and detail more specifically how we involved the prospective users in the development process.

### 4.1 Embracing the guidelines of interface design

For the design of the interface to our Bayesian network for classical swine fever, we studied the data-entry task to be performed by its prospective users. The task itself basically is an observation-and-report task: a veterinarian has to examine a pig and to report his clinical findings as data items. The task moreover is unambiguously defined by the details of the network. The network's variables define the nominal categories to which the data items to be reported belong; for each data item moreover, a complete list of possible options is available, which correspond with the values of the variable at hand. For this type of observation-and-report task, questionnaires with closed questions are known to constitute a suitable construct for data entry [5]. By embracing this construct, giving evidence for our Bayesian network is re-framed as filling in a questionnaire. The *natural-language* and *user-compatibility* guidelines reviewed above now recommend that this questionnaire uses simple, natural language, preferably stated in veterinary terms.

The small screen of the personal digital assistant on which the interface to our Bayesian network is to be run, clearly forestalled the use of fully-phrased questions and answers as support for the data-entry task. The size of the screen in fact necessitated the use of keywords to refer to the network's variables and associated values. These variables and values had been given names by the network developers, based upon the insights they had gained from the interviews held for the network's construction. Following the *user-compatibility* guideline, we decided however not to use these names for the data-entry interface, but to elicit appropriate keywords directly from the prospective users themselves; we will elaborate on the elicitation meeting held for this purpose in Section 4.2. The use of keywords instead of full questions would clearly carry the risk of compromising the guideline of *preserving precision* formulated in Section 3: even well-chosen keywords allow some freedom of interpretation. To guard the mathematically precise meaning of a variable or value therefore, we decided to offer a user the possibility of consulting the full question by touching an information button marked '?' to the right of each keyword; an example of the message box which then pops up, is shown in Figure 3. By providing this possibility all relevant information was in essence preserved.

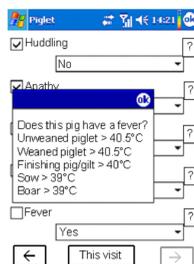


Figure 2: The example message box on the pda screen, containing a full question and its possible answers

As a side effect of the small screen of the personal digital assistant, the keywords format significantly shortened reading time upon data entry. Taking the *efficient-system* guideline even further, we decided to show a default answer in the pull-down answer box for each question. If a user does not wish to accept this default answer, he has to select an appropriate answer from among the list of possible options provided in the answer box; following the action, a tickmark appears in an 'ok' button to the left of the question. If the user does want to accept the default answer, he has to tick this 'ok' button himself. Since it is important that each question be looked at and answered consciously, we decided to explicitly forestall automated and possibly unintentional acceptance of a default answer by requiring the user to perform an action. We note that requiring an explicit action is in line with the *preserving-precision* guideline.

To conclude, we aligned the order of the requests for clinical information to the workflow of veterinarians upon visiting a pig farm, thereby once more following the *user-compatibility* guideline for interface design.

## 4.2 The elicitation of keywords for data entry

In the previous section we reviewed several considerations about the data-entry interface to our Bayesian network for classical swine fever. To follow up on these considerations, we had to construct a list of keywords, stated in veterinary terms, to describe the various data-entry questions. Similarly, appropriate keywords had to be chosen for the possible answers to these questions, and default answers had to be selected. Since the screen of the personal digital assistant allowed showing just a limited number of questions at a time, we further had to distribute the various questions over multiple screens in a way that matched the veterinarians' workflow. For eliciting all required information, we organised a meeting with twelve veterinary practitioners who were considered representative of our network's future users. At this meeting, we demonstrated a prototype version of the data-entry interface, stated in the developers' terms, and explained our goals. After this introduction, the participants were assigned to two groups of six practitioners each. One group was asked to suggest keywords and default answers for all data-entry questions (task 1), and the other group was requested to cluster the questions and to order the resulting clusters along their workflow (task 2).

### *Task 1: Finding keywords and default answers*

The main goal of task 1 was to arrive at a collection of keywords, stated in veterinary terms, to describe the various questions for our data-entry interface; we further wanted to identify a default answer to each question. To this end, we designed a paper form containing full phrases in natural language for each variable and each value from our Bayesian network. Upon constructing the various phrases, we noticed that a positive answer to a question sometimes indicated the presence of some clinical sign (as for example in *Conjunctivitis* = yes), while it indicated the absence of a sign for other questions (*Activity* = normal). Since having to switch the direction of reasoning upon interpreting questions can be quite confusing in practice, we decided to rephrase all questions into the same reasoning direction (*Apathy* = no, rather than *Activity* = normal). Whenever possible, we further rephrased the answers to the questions to simply 'yes' and 'no'. We would like to note that these changes already resulted in a much more consistent data-entry interface than a literal translation of the names of the various variables and values had done. The six veterinarians participating in the task were now handed the constructed paper form, and were asked to think of one or a few keywords to characterise each question and to write these keywords down on the form as indicated. As they had just seen the prototype version of the data-entry interface, they knew approximately how many characters would fit on the screen of the hardware. Next to each question, the paper form stated the possible answers. The participants were asked to choose the answer that they expected to give the most frequently for a question, when using the personal digital assistant on farms with clinical problems of unknown origin. The indicated answer was to become the default answer to the question on the screen of the data-entry tool. All in all, the participants were asked to generate 26 keywords and to identify 26 default answers.

After the meeting, we studied the keywords and default answers provided by the six participants on the paper forms. For each data-entry question, we identified the keyword, or combination of keywords (possibly constrained by the usable space on the screen of the hardware), that was given the most often. Interestingly, less than 30% of the thus established keywords matched the names of the variables in our Bayesian network. Although these variable names had not been chosen specifically with the intention to be used in the data-entry interface, this small percentage illustrates quite nicely how much developer-provided names can differ from the keywords given by projected end-users. Even when carefully choosing keywords for the prototype interface to be demonstrated during the elicitation meeting, didn't the network developers do significantly better: the agreement of their keywords with the user-provided ones increased by 5% only.

From the filled-in paper forms, we further established for each data-entry question, the answer that was indicated the most often by the veterinarians as the default answer. For the six questions for which the result was undecided, we selected from among the most frequently indicated answers, the one that had highest prior probability according to our Bayesian network. We also compared the list of default answers provided by each individual veterinarian to the values with highest probability established from the network. To our surprise, we found large differences in the percentages of agreement, which ranged from 30% to 70%. To investigate the origin of these differences, we entered into our network the various possible diagnoses (no infection, a primary respiratory infection, a primary intestinal infection, and an infection with the classical swine fever virus) and compared the posterior probability distributions computed for the clinical variables against the default answers given by the veterinarian; reversely, we entered the provided default answers and established the probability distributions over the possible diagnoses given these answers. We found that one of the participants clearly had had a healthy pig without any infection in mind when selecting the default answers. Two veterinarians had thought of a pig with a CSF infection, which was not entirely surprising as the meeting had been announced as pertaining to classical swine fever. The pattern of clinical signs typically seen in a pig with a respiratory infection was what two other veterinarians had had in mind, and one veterinarian had actually envisioned a pig with both a gastrointestinal and a respiratory infection. These findings suggest that giving default answers based on an essentially averaged clinical pattern of disease, was a challenging task for the practitioners: apparently they had tried to envision the task in their professional setting, in which no pig with an averaged clinical pattern would exist. Another interesting finding from studying the provided default answers, was an evident misunderstanding by the veterinarians of one of our questions, despite its phrasing in simple natural language. For the question 'Is this pig barn very dusty?', all participants had indicated that the most frequently used answer would be 'yes'. We had meant to ask whether the pig barn was so dusty that the level of dust could offer an explanation for an observed pattern of clinical signs in the barn. Without this addition, the veterinarians had indicated that in fact most pig barns are quite dusty. This finding serves to show that simple misunderstandings are bound to arise, even if a system's developers are fully aware of differences in language and in mental model with the system's prospective users.

#### *Task 2: Finding natural clusters of questions*

The goal of task 2 was to establish a clustering of the various data-entry questions, such that the questions from a single cluster were closely interconnected and the questions from different clusters were less strongly related, according to the participants. In the data-entry interface, the clusters of questions would be shown on separate screens. Since the question screens would be presented consecutively, we further wanted to find an ordering of the clusters which reflected the practical workflow of veterinary practitioners. We observed that the clustering task in essence was a categorisation task. A suitable method for eliciting categorisations is the use of concept sorting: sorting concepts is a natural reasoning task, as the human brain tends to sort related concepts into categories [6]. To offer to our participants some visual support for the sorting task to be performed, we constructed a set of 26 paper cards, each containing a summary keyword and the matching full question and full answer options. We would like to note that, since the two tasks described in this section were performed in parallel, the keywords from task 1 were not available as yet for use with task 2; the summary keywords on the cards were therefore keywords couched by the developers. Each of the six participants now received the set of cards, along with a written instruction for the clustering task to be conducted. Each participant was also asked to characterise each constructed cluster with a collective name, to be written down on a separate blank card, and to add this card to the cluster. The additional card was intended for the developers, for insight into the criteria by which the cluster was constructed. The final task of the participants was to put their clusters of questions in the order in which they would like to answer them upon visiting a pig farm. Additional blank cards were provided with the set of 26 pre-printed cards, for the veterinarians to write down any relevant questions that they considered missing from the set.

The veterinarians constructed 6 to 9 question clusters each. The number of cards per cluster ranged from 1 to 16; the mean cluster size was 4.3 including the originally blank cards with added questions, and 4.0 including the pre-printed cards only. The clusterings provided by the six participants were summarised in a  $2 \times 2$  matrix of the frequencies with which two questions were included in the same cluster. From this matrix, we identified clusters of questions with maximized frequency and with at most five questions each. Upon studying the cluster orderings provided by the six participants, we found two essentially divergent types. Both types of ordering started with a cluster of questions which an attending veterinarian would ask of the farmer. The first type then proceeded by ordering the clusters from questions which could be answered by observing the pigs and their interactions from a distance, to questions which could only be answered by

closer clinical examination of individual pigs. In the second type of ordering, the farmer questions were followed by clusters of questions related to different types of clinical problem, such as gastrointestinal or respiratory problems. Apparently, the veterinarians had adopted different strategies for ordering the clusters of questions: while the first type of ordering appears to have its origin in a veterinarian's usual workflow, the second type resembles the way in which textbooks describe clinical patterns of disease. Since divergent strategies were found for ordering the clusters of questions, we should probably allow users a choice in strategy for our data-entry interface. Because programming such a choice on the hardware to be used would be quite involved, we decided upon a fixed ordering of clusters for the field-trial version of our interface.

## 5 A Preliminary Evaluation of the Data-entry Interface

The data-entry interface to our Bayesian network for classical swine fever was programmed on a personal digital assistant, following the design considerations described above. Eleven veterinarians were supplied with the resulting tool, for use in a field trial of 22 months. At the start of the trial, the participants received a training to make them acquainted with the full questions underlying the keywords used in the interface.

After six months in the trial, we held brief interviews by telephone with two of the participants. The purpose of these interviews was to obtain early feedback on the ease of use of the data-entry interface on site at pig farms. Although they had experienced some practical problems with the tool's hardware, both veterinarians were quite enthusiastic about the interface itself: they found it 'logical and valuable'. Still, they had to get used to working with the tool in a barn: 'you do have your hands full now!'. Both veterinarians found most questions quite clear, except for two questions pertaining to the effect of antibiotic treatment and to the presence of a fever, respectively. With respect to antibiotic treatment, one of the veterinarians remarked that judging whether or not clinical signs had diminished following treatment, was not always easy: he was not sure what to answer if a pig had relapsed after a seeming recovery for a single day. The other veterinarian commented that the different temperature thresholds used for defining a fever for the different pig types, were hard to memorise: he consulted the full question and associated answers quite frequently, to remind him of the precise numbers. Both veterinarians further mentioned that they often used the option to accept the default answer. Since all touch strokes from all participants had been logged, we evaluated the use of the default answers in the data received so far. The results indicated that for 80% of the questions, the default answer was the most frequently given. We also asked the two veterinarians whether using the data-entry tool had altered their task execution. One of them indicated that, if it had not been for the field trial, he would not have measured the pig's temperature in all cases. Both practitioners stated that they would normally collect the same information, but that they now looked more consciously. We concluded, cautiously, that our data-entry interface fits the veterinarians' best practices.

After the field trial had finished, we organised a more involved evaluation meeting, which was attended by eight of the eleven veterinarians who had participated. For this meeting, we prepared a written questionnaire with closed questions about the ease of use of the data-entry tool for our Bayesian network. The reports obtained basically served to confirm the findings from the earlier informal evaluation, and supported more specifically the cautious conclusion that our design efforts had resulted in a practicable interface. There were some complaints about technical aspects of the hardware, but there were hardly any negative remarks about the interface itself. The veterinarians had all used the data-entry tool and had continued to use it throughout the trial, except for one veterinarian who had difficulties reading the information on the tool's small screen when in a barn. Entering the requested clinical information took the veterinarians some extra 6 minutes per pig on average. And, although the interface forced a specific order of data entry, most veterinarians found the imposed workflow quite natural, or else could easily adapt to it.

For investigating whether we had been able to preserve the precision of the definitions of the variables and values from our Bayesian network, we prepared a multiple-choice questionnaire which listed the 26 keywords used in the data-entry interface. With most of the keywords, three possible definition options were provided. One of these options was the correct definition from our network, one option was less specific than the correct definition, and the remaining option was more specific. The participating veterinarians were asked to indicate the correct definition from among these three options, without consulting the interface on the personal digital assistant. For two of the 26 keywords, the questionnaire included an open question, in answer of which the veterinarians had to provide one or more numbers. The results from the questionnaire showed that for 65% of the keywords, the precision had been preserved, in the sense that at least seven of the eight participants had indicated the correct definition. For 17% of the keywords, the correct definition had been selected by five or six veterinarians; for 18% of the keywords, fewer than five participants had identified

the correct definition. In 98% of all incorrect answers, the veterinarian had chosen the less specific definition option. The keywords with the largest percentages of incorrect definitions, were the two keywords for which the veterinarians had to provide one or more numbers. Clearly, precision is much harder to preserve for concepts with a complex compound definition than for concepts with a relatively straightforward meaning.

## 6 Conclusions

After having developed a Bayesian network for the early detection of classical swine fever in pigs, we addressed the design of an appropriate data-entry interface for its prospective users. The interface had to be tailored to on-site use by veterinary practitioners upon visiting a pig farm with disease problems of unknown origin. Since the intended use implied demands of portability and ease of disinfection, we decided to implement the data-entry interface on a personal digital assistant. For the design of the interface itself, we embraced four well-known guidelines recommending for example the use of simple, natural language and alignment with the usual workflow of projected users. From these guidelines, we concluded for example that the commonly-offered interface of modern Bayesian-network tools does not constitute a suitable interface for our veterinary users. While the four design guidelines proved highly valuable for our context, we noticed that none of them sufficed for guaranteeing that the mathematically precise definitions of the stochastic variables of our Bayesian network were retained throughout all user interactions. We proposed a new guideline for this purpose, which we coined *preserving precision*. The new guideline pertains especially to the design of interfaces for non-mathematical users to automated systems which build upon mathematically precise concepts and for which mathematical precision needs to be retained. We demonstrated how we used the new guideline upon developing the data-entry interface for our Bayesian network for classical swine fever and how we solicited the help of the prospective users in its design.

The newly designed data-entry interface to our Bayesian network was tested in a field trial of 22 months, during which it was used by eleven pig veterinarians in their daily practice. After the trial had ended, evaluation of its ease of use showed that our design efforts had resulted in a practicable interface, about which the participating veterinarians had hardly any negative comments. Our evaluation of the extent to which we had succeeded in preserving precision, showed that the veterinarians had retained the correct definitions for the majority of the stochastic variables for which they had to enter clinical data. The evaluation results also revealed however, that definitions of the more complex variables with compound meanings were less well preserved. Our further research will now focus on the formulation of design heuristics bearing on our new guideline, including for example heuristics for better retaining compound mathematical definitions. When supplemented with such heuristics, we feel that our guideline of *preserving precision* has the potential to become an integrated part of best practices of interface design.

**Acknowledgments.** This research was supported in part by the Central Veterinary Institute of Wageningen UR. We are most grateful to the veterinarians who participated in our study, for their invaluable input. We further thank Petra Geenen for her help with the elicitation and evaluation meetings, and Martijn Schrage and Arjan van IJendoorn for their programming efforts.

## References

- [1] J. Nielsen (1993). *Usability Engineering*. Academic Press, San Diego, CA.
- [2] D.J. Mayhew (1992). *Principles and Guidelines in Software User Interface Design*. Prentice Hall, Englewood Cliffs, NJ.
- [3] B. Togazzini (downloaded 2006). <http://www.asktog.com/basics/firstPrinciples.html>.
- [4] L.C. van der Gaag, E.M. Helsen (2002). Experiences with modelling issues in building probabilistic networks. In: A. Gomez-Perez, V.R. Benjamins (editors). *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, pp. 21 – 26. Springer-Verlag, Berlin.
- [5] J.A. Gliner, G.A. Morgan (2000). *Research Methods in Applied Settings: an Integrated Approach to Design and Analysis*. Lawrence Erlbaum Assoc., Mahwah, NJ.
- [6] J.R. Anderson (2004). *Cognitive Psychology and its Implications* (6th edition). Worth Publishers, New York, NY.

# Designing a Search and Rescue Simulation Environment for Studying the Performance of Agent Organizations

Mattijs Ghijsen <sup>a</sup>

Wouter Jansweijer <sup>a</sup>

Bob Wielinga <sup>a</sup>

<sup>a</sup> *System and Network Engineering Group,  
Informatics Institute, University of Amsterdam*

## Abstract

In the study on performance of organizations of Multi-Agent Systems there exists a need to understand the effects of the task-environment and organization of the agents on the performance of Multi-Agent Systems. Current simulation environments often lack sufficient control over the environment and lack the ability to systematically vary a number of task-environment and organizational parameters and measure the effect of these changes on performance. For this purpose we have created the Extended Organization Design model which categorizes and describes aspects of Multi-Agent Systems; their organization, their task-environment, and a set of performance metrics. We show how the Extended Organization Design is used as a basis for a parameterized model of the Search and Rescue domain.

## 1 Introduction

For agents in a Multi-Agent System (MAS) to cooperate effectively and efficiently, organization is required. To study and understand the performance of such organizations, simulation tools can be used. In this paper we address the issue of constructing simulation environments that allow for a systematic analysis of multi-agent organization performance. We have chosen Search and Rescue (S&R) as an application domain due to its challenges for MAS research such as its distributed and cooperative nature and high degree of uncertainty and dynamics.

A number of simulation platforms have already been developed for simulations in the disaster management domain. Well known is the RoboCup Rescue simulation system (RCRSS) [1] which aims at comparing the performance of a number of different MAS organizations in exactly the same setting. However it lacks easy manipulation of the task-environment. Other simulation environments such as the Urban Search And Rescue simulator (USARsim) and the distributed building evacuation simulator [2, 3] also provide realistic simulation environments but lack sufficient control to manipulate the task environment.

An example of a simulation environment that provides more control to the user is the predator-prey pursuit simulation system [4]. In this system, an explicit mathematic model is provided to describe the predator prey domain. Another example of a more controllable environment is a system for simulating software evolution [5]. Although no explicit environment model is presented and the application domain is completely different from ours, their simulator allows for the systematic variation of a number of parameters and the authors use a clear methodological approach to analyze the results. The latter is a good example of the type of experiments we envision for our simulation environment.

So and Durfee [6, 7] present a more systematic on studying MAS performance. More specifically they present an organization design model in which the performance of a MAS is influenced by the task-environment and the organizational factors of the MAS. Moreover, they recognize that interaction effects exist between the task-environment and MAS organization factors. Their model however does not provide specific task-environment factors and MAS organization factors. Virginia Dignum [8] and Frank Dignum [9] present their approach for the design of a simulation tool for studying MAS reorganization. They first identify the factors that determine the need for organization. Then they explore the different ways of reorgani-

zation and finally they identify the different triggers for reorganization. Based on this generic framework for reorganization a simulation environment for reorganization is defined. In our approach we combine the basic framework presented by So and Durfee with the design approach by Dignum et al. to describe a methodology for designing MAS simulation environments that can be used for a systematic analysis of MAS performance.

In this paper we present a methodology that consists of a theoretical framework, the Extended Organization Design (EOD) model, and an approach for using the EOD to design a simulation environment. In Section 2 we discuss the EOD model which is based on the organization design model by So and Durfee. The EOD extends the organization design model with a vocabulary to describe the MAS organization and the task-environment in which the MAS organization operates. Furthermore, we provide a more detailed performance model that distinguishes between effectiveness and efficiency and provide a set of performance metrics. In Section 3 we demonstrate our approach to operationalize the generic factors of the EOD for a Search and Rescue Simulation environment. Next, in Section 4 we show how the simulation environment is used in an experiment to analyze the impact of communication failure on the performance of a MAS organization.

## 2 Extended Organization Design Model

The model of organization design by So and Durfee [7] explains the interaction between a MAS organization and its task-environment and the effect of this interaction on the performance of a MAS.

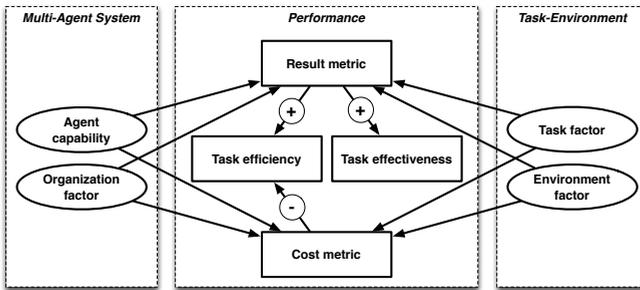


Figure 1: The Extended Organization Design Model.

In this paper we extend the organization design model with a vocabulary for describing a MAS organization, its task-environment and MAS performance metrics. In the latter we identify a number of result oriented and cost oriented metrics. Another extension of the organization design is a more detailed performance model in which we distinguish between effectiveness and efficiency and we couple these concepts to the result and cost oriented performance metrics. Our Extended Organization Design (EOD) model is shown in Figure 1.

### 2.1 EOD Performance model

```

Performance metric
|--- Result Metric
|   |--- Solution quality
|   |--- Time-to-goal-achievement
|--- Cost Metric
|   |--- Resource consumption
|   |--- Communication costs
    
```

Figure 2: Performance metrics

The EOD performance model distinguishes between result oriented and cost oriented metrics. Based on these two types of metrics, we define effectiveness as the ratio between the obtained result while trying to

achieve a goal or performing a task and the maximum obtainable result. Efficiency is defined as the ration between the obtained results and the costs that have been made while trying to achieve a goal or performing a task. Figure 2 shows the different performance metrics in the EOD model. We distinguish between the four types of performance metrics described in [10], solution quality, time-to-goal-achievement, resource consumption and communication-costs.

## 2.2 EOD Task-Environment Model

```

Task factor
|--- Task size
|--- Task complexity
|       |--- Task decomposability
|       |--- Subtask heterogeneity
|       |--- Inter-subtask relations
|       |--- Subtask distribution
|--- Task reward
|--- Task dynamics

```

Figure 3: Task factors

Figure 3 shows the different task factors of the EOD model. We distinguish between the size of the task, factors that determine the complexity of the task, the reward that can be received by performing the task and the task dynamics. Task size relates to the amount of work that needs to be performed. Task complexity describes how easily a task can be composed into subtasks, the heterogeneity of the subtasks and the relations between subtasks.

The reward of a task describes the amount of reward that can be obtained by an agent or its organization if a task is performed. Dynamics in the task size, complexity or reward, require the organization and agents in the organization to constantly adjust their planning and may also lead to more uncertainty in the organization when agents are not able to keep up with dynamics in their tasks.

```

Environment factor
|--- Communication factor
|       |--- Capacity
|       |--- Reliability
|--- Resource factor
|       |--- Scarceness
|       |--- Distribution
|       |--- Types
|--- Topology factor
|       |--- Size
|       |--- Accessibility
|--- Behavior factor
|       |--- Observability
|       |--- Determinism
|       |--- Dynamics

```

Figure 4: Environment factors

The EOD’s environment factors are shown in Figure 4. The first factor shown is *communication* and we identify the capacity and reliability of the communication infrastructure as its two main aspects. The next factor is the resource factor. Resources can be described in terms of their scarceness, how they are distributed over the environment and the different types of resources (e.g. consumable or reusable). The third factor is the topology factor of the environment. This factor is defines the size of the environment and the accessibility of the environment. Finally, we define a number of behavior factors of the environment. The observability of the environment – which can be full or partial – is related to whether or not relevant information for decision making can be observed by the agents in the environment. The determinism factor indicates if the outcome of an agent action in a certain state will always result in the same next state or not. Dynamics in the environment determine how the environment changes “spontaneously” without any agent action causing the change.

## 2.3 EOD Multi-Agent System Model

```

Agent factor
|--- Physical capability
|--- Knowledge
      |--- Declarative knowledge
      |--- Procedural knowledge

```

Figure 5: Agent factors

The agent factors, shown in Figure 5, consist of two main aspects: the physical capabilities and the agents knowledge. The physical capabilities determine how the agent interacts with its environment, how (well) the agent observes its environment and which actions is the agent able to perform on the environment. The knowledge of an agent is consists of declarative knowledge and procedural knowledge.

```

Organization factor
|--- Organization size
|--- Organization heterogeneity
|--- Organization structure
      |--- Communication structure
      |--- Normative structure
      |--- Social structure
      |--- Interaction structure

```

Figure 6: Organization factors

The organization factors, shown in Figure6 consists of three main aspects: the size, the agents that form the organization and the structure of the organization. Organization size determines the amount of work that could potentially be done by an organization. We define the heterogeneity of a MAS organization by the heterogeneity of the agents that form the organization. Agents may have different physical capabilities as well as different knowledge. Following [11], we identify the following structural factors of a MAS organization: the communication structure, the normative structure, the social structure and the interaction structure.

## 2.4 Using the EOD model

The EOD model in this section provides a framework for the designer of a MAS environment. Because the covers a wide range of factors, it helps the designer to be explicit about the design choices that are made. By being explicit about the design choices, the designer provides more insight in the environment to the user.

To implement of the EOD factors in a specific domain we identify two steps. The first is the operationalization step in which an EOD factor is represented by a more specific concept. For example the task size factor can be operationalized as the number of actions that need to be performed in order to complete the task. The second step is the implementation step in which an operationalized factor is parameterized. For example, in the Search and Rescue domain, the number of actions to complete a task can be implemented by two parameters: the number of victims that need to be rescued and the size of the search area.

## 3 Environment Design

In applying the EOD model to design a search and rescue simulation environment we have to operationalize and implement the EOD factors. To balance the amount of realism and the needed simplicity for a controllable environment we often had to apply two operationalization steps at once. First we introduce search and rescue factors to operationalize the EOD factors. At the same time these operationalized S&R factors are often also a simplification of the real world search and rescue domain. The goal of this simulation environment is to provide a parameterized simulation environment that allows for systematic variation of (mainly) task-environment and (partly) MAS organization parameters. It is not the intention to provide a complete instantiation of the EOD model. A complete implementation of the EOD model, i.e. at least one parameter for each task-environment and organization factor and at least one performance metric for each of the generic performance metrics, is out of the scope of this research.

The S&R simulator is a discrete-time simulator. The main motivation for discrete-time is that this makes it easier to implement a system with reproducible results. The environment consists of a rectangular grid topology on which a number of victims are distributed. We have chosen for a rectangular grid to limit agent movements to just 4 directions and make the speed in which the agents move around more controllable. The victims have a certain health status which may decline over time. The initial health state and the decline of health represents how serious a victim is injured. Victims have a fixed location and cannot move. In order to rescue a victim, agents first have to find the victim and then cooperate to rescue the victim by jointly performing a rescue action in the same time-step. In order to find victims, agents can move around on the grid. In a single time-step, an agent can move either one grid cell up, down, left or right. When an agent is moving around, it is able to observe the grid cells that are within its viewing range. These observations are always accurate. Actions related to rescuing a victim and moving around the search area are deterministic. In order to cooperate, agents may need to communicate with each other. To facilitate communication, the simulator provides the agents with a wireless communication infrastructure. Actions related to communication are non-deterministic due to possible failures in the communication infrastructure. The simulator environment is partly observable, i.e. agents cannot see all relevant information needed for their decision making. For example, agents cannot observe whether a communication tower is operational or not.

The communication infrastructure is a simplified wireless communication network which covers the complete search area. Whenever an agent sends out a message, that message is picked up and sent to the receiver(s). Three types of messages are available to the agent; unicast, multicast and broadcast messages. A broadcast message is sent to all agents on the search area. In the case a directed (unicast or multicast) message is used, the sender has to specify one or more receivers of that message. Each time-step, an agent is allowed to send one message and the message size is limited. The reliability of the network is determined by the uptime of the network. Whether the network is up or down is determined each discrete time step with a probability ranging between 0% and 100%.

The physical capabilities of an agent are defined by its viewing range and the maximum amount of messages the agent can receive per time-step. The simulator does not impose any constraints and does not influence any of the knowledge factors of an agent.

For the Search and Rescue domain, many different types of performance metrics are possible. In our simulator we support three result metrics: the total reward that is received (i.e. the summed health of all victims at the end of a simulation), the amount of victims that are rescued and the amount of time taken to rescue all victims. Furthermore, the simulator supports two cost metrics that both focus on communication-costs: the amount of bytes that are sent and the amount of bytes received by agents.

## 4 Evaluation

To demonstrate the use of the simulator, we describe a case-study on the performance of a MAS organization. First we describe the MAS organization that coordinates using mutual adjustment. Next, we describe the design, data gathering and analysis of the performance evaluation study on the influence of network reliability and workload on performance.

### 4.1 Organization Design

The organization that we have designed uses a coordination mechanism that can be characterized as mutual adjustment [12]. This means the agents form a decentralized organization in which agents mutually adjust their actions to each other in order to perform their tasks. The interaction mechanism that is used is similar to a Contract Net [13]. The Contract Net provides a generic mechanism for communicating bids for cooperating on a task, the content of those bids and the offers that other agents can send.

In this organization, an agent can rescue a victim in two ways: the agent can decide to form a coalition for rescuing the victim, or the agent can decide to join a coalition. Forming a coalition consists of the following steps: The agent sends a request for forming a temporary coalition. Other agents can respond to this request by sending an offer to join the coalition. If the coalition accepts the offer, the coalition is formed and the agents will rescue the victim at the agreed time.

The messages that are used in this interaction are `<request>`, `<offer>` and `<accept>` and the content of these messages is shown in Figure 7. A `<request>` message consists of an expiration time which indicates how long the request is valid, an action-window which is the time-window in which the

```

<content> ::= <request> | <offer> | <accept>;
<request> ::= exp-time, action-min, action-max, victim-x, victim-y;
<offer>    ::= exp-time, action-min, action-max;
<accept>  ::= rescue-time;

```

Figure 7: Messages types and content.

action should take place and the coordinates of the victim. When the expiration time (`exp-time`) time has expired, the sender and receivers of this message will no longer consider this offer. The action-window allows other agents to decide if they will be able to join the coalition in time at the given location of the victim. If the agent decides to join a coalition, the agent will send an offer with a limited expiration time and the agent indicates its availability by an action-window which is a subset of the action-window in the `request` message. When the requesting agent has received sufficient offers, the agent will then send the `accept` message to the agents that will form the coalition. This `accept` message contains the time-step in which the rescue action should take place.

In this communication scheme, `request` messages are broadcasted while `offer` and `accept` messages are directed messages (unicast and multicast respectively). Furthermore, to prevent agents from flooding the communication infrastructure by broadcasting requests, each agent is only allowed to have one valid outstanding request.

## 4.2 Evaluation Setup

The goal of this evaluation is to study how the reliability of the communication infrastructure affects the solution quality performance of the aforementioned MAS organization. When the network uptime is less than 100%, two types of events can occur in the MAS organization's interaction pattern due to communication failure. First, when the communication network is down for one or only a few time-steps, agents are still able to respond to each others messages before these messages expire. When the communication network is down for longer periods of time, agents will not be able to respond to messages before they expire.

Based on these two delays we hypothesize that when the uptime of the network decreases, the first type of delay will start to occur in the MAS organizations's interaction pattern and performance will drop. Then, when we further decrease the uptime of the network, the second type of delay will also start to occur. This will cause a more severe drop in performance. Once the uptime of the network reaches 0%, performance will also drop to 0.

## 4.3 Results

Data for this evaluation was gathered by varying the network uptime between 0% and 100% with a 2% step size. Each simulation was done on a  $30 \times 30$  search area. The number of victims was varied from 15, to 60 to 120 and the initial health state of a victim was set to 100 and the health decreased with 0.2 every time-step. Furthermore, the organization consisted of 30 S&R agents, each with the same observability range ( $5 \times 5$  range) and the same receive capacity of 100 messages per time-step. Each simulation is initialized with a different random seed which causes a different distribution of victims, different initial agent positions, and different timing of network failure.

The results of the simulations are shown in Figure 8. We measure the effectiveness by measuring two performance metrics, the total victim health at the end of a simulation and the number of victims rescued during a simulation. For the first performance metric, effectiveness is obtained by dividing the total victim health at the end of the simulation with the total initial health of all victims. For the second performance metric, effectiveness is obtained by dividing the total number of victims that are rescued at the end of the simulation by the total number of victims in the simulation.

The results show the expected decrease in effectiveness for victim health and number of victims rescued when the network uptime decreases.

When we look at the total victim health, it shows that the initial decrease in effectiveness is relatively slow. This can be explained because at high uptime values, communication failures mostly cause small delays. At a certain point however, larger delays are caused by larger periods of network downtime. When the workload per agent is relatively low, the agents still manages to rescue a lot of victims despite the network

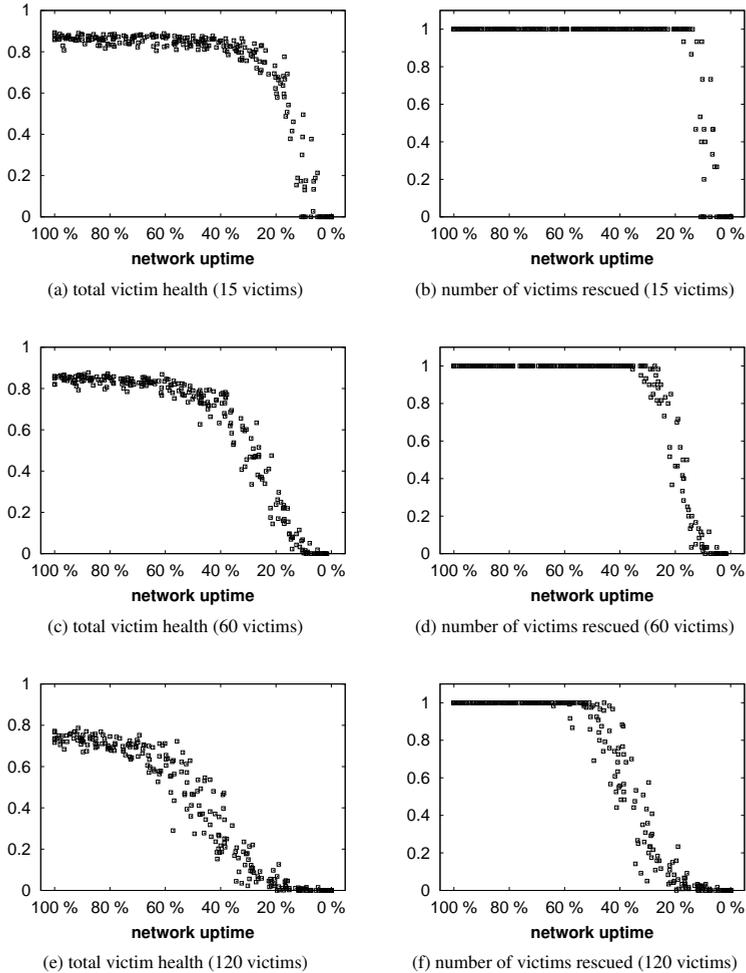


Figure 8: Influence of network uptime on effectiveness for total victim health and number of victims rescued.

downtime. However as the workload increases, the delays caused by network downtime prevent the agents from rescuing victims quickly and their total health decreases.

Furthermore, when we look at the number of victims being rescued, it is clear that the uptime of the network influences the maximum number of victims that can be rescued. 15 victims can still be rescued when the uptime is around 15%, 60 victims can be rescued when the uptime is around 30%, while 120 victims can still be rescued when the uptime is around 50%. This indicates a non-linear relation between the uptime and the number of victims that can be rescued.

## 5 Conclusions

In this paper we present a methodology for the systematic design of simulation environments. Our methodology consists of the Extended Organization Design model which is a domain-independent model to describe organizations of agents, the task-environment in which they operate and how performance is influenced by task-environment and organization factors. The EOD model provides a structure of and vocabulary for task-environment factors, MAS organization factors and performance metrics. Furthermore, we provide an two step approach for implementing the EOD factors as parameters in a simulation environment.

We have used our methodology to design an agent simulation environment for the Search and Rescue domain. The main aim was to create a controllable experimentation environment for conducting experiments on the performance of Multi-Agent Organizations. In an experiment we show how the simulation environment is used to analyze the effect of communication failure with different levels of workload on the performance of a MAS organization. In this analysis we use two complementary measures of effectiveness to understand the agents behavior when communication is failing.

## References

- [1] Hiroaki Kitano, Satoshi Tadokoro, Itsuki Noda, Hitoshi Matsubara, Tomoichi Takahashi, Atsushi Shinjō, and Susumu Shimada. RoboCup-Rescue: Search and Rescue for Large Scale Disasters as a Domain for Multi-Agent Research. In *Proceedings of IEEE Conference on Man, Systems, and Cybernetics (SMC-99)*, pages 739–743, 1999.
- [2] Nikolaos Dimakis, Avgoustinos Filippoupolitis, and E. Gelenbe. Distributed Building Evacuation Simulator for Smart Emergency Management. *The Computer Journal*, 53(9):1384–1400, February 2010.
- [3] Avgoustinos Filippoupolitis and Erol Gelenbe. A distributed decision support system for Building Evacuation. In *2009 2nd Conference on Human System Interactions*, pages 323–330. Ieee, May 2009.
- [4] Javier A Alcazar and Epharahim Garcia. Parametric Analysis for Modeling and Simulation of Stochastic Behavior in the Predator Prey Pursuit Domain. *Simulation*, 82(12):827–840, December 2006.
- [5] Benjamin Stopford and Steve Counsell. A Framework for the simulation of Structural Software Evolution. *ACM Transactions on Modeling and Computer Simulation*, 18(4):17–45, September 2008.
- [6] Young-pa So and Edmund H Durfee. Designing Tree-Structured Organizations for Computational Agents. *Computational and Mathematical Organization Theory*, 2(3):219–246, 1996.
- [7] Young-pa So and Edmund H Durfee. Designing Organizations for Computational Agents. In M Prietula, K Carley, and L Gasser, editors, *Simulating Organizations*, pages 47–64. AAAI Press/ MIT Press, 1998.
- [8] Virginia Dignum, Frank Dignum, Vasco Furtado, Adriano Melo, and Liz Sonenberg. Towards a Simulation Tool for Evaluating Dynamic Reorganization of Agents Societies. In *Workshop on Socially Inspired Computing. AISB Convention 2005.*, 2005.
- [9] Frank Dignum, Virginia Dignum, and Liz Sonenberg. Exploring Congruence Between Organizational Structure and Task Performance: A Simulation Approach. In *AAMAS Workshops*, pages 213–230, 2005.
- [10] Mattijs Ghijsen, Wouter N H Jansweijer, and Bob J Wielinga. Agent Decision Making for Dynamic Selection of Coordination Mechanisms. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 87–91, 2008.
- [11] Virginia Dignum. *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. PhD thesis, Universiteit Utrecht, 2003.
- [12] H Mintzberg. *Structures in fives: designing effective organizations*. Prentice Hall, Englewood Cliffs, N.J., 1993.
- [13] R Smith. The contract net protocol: high level communication and control in distributed problem solver. *IEEE Transactions on Computers*, 29:1104–1113, 1980.

# A Generalization of the Winkler Extension and its Application for Ontology Mapping

Maurice Hermans

Frederik C. Schadd

*Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands*

## Abstract

Mapping ontologies is a crucial process when facilitating system interoperability and information exchange. Ontology Mapping systems commonly utilize string metrics in the mapping process to compare concept names. String metrics can be extended using the Winkler method, which increases the similarity value of two strings if these have a common prefix. A common occurrence for two corresponding ontology concepts is that the name of the first concept is a non-prefix sub-string of the name of the second concept. The Winkler extension does not allocate a higher similarity value to these pairs of strings, however intuitively this indicates that the two names have a similar meaning. This paper proposes a generalization of the Winkler extension, such that pairs of names with large common non-prefix sub-strings receive a higher similarity value as well. The proposed metric is evaluated on a record-matching dataset and a dataset from the Ontology Alignment Evaluation Initiative. The experiments reveal that metrics applying our proposed generalization outperform the same metrics when applying the Winkler extension.

## 1 Introduction

Ontologies commonly form the basis of modern knowledge systems. These ontologies are created by domain experts to suit the needs of the specific knowledge system. Hence, it is likely that two ontologies describing the same domain, but originating from different knowledge systems, will contain differences such as heterogeneous concept names, structure or granularity. Facilitating information exchange between knowledge systems which are based on heterogeneous ontologies is a challenging, but crucial task. In order to exchange information between two ontologies, a mapping is required which identifies the correspondences between the ontology concepts. The task of matching is a critical operation in many fields, such as semantic web, schema/ontology integration, data warehouses, e-commerce etc. While contemporary knowledge systems are commonly based on ontologies, the problem of mapping conceptualizations of knowledge domains originates from the field of databases, which utilize schemas to encode meta data.

The task of matching takes as input two ontologies, each consisting of a set of concepts and determines as output the relationship. There are multiple relationships possible e.g. equivalence, subsumption but in this article we only deal with equivalence. To match two concepts there are numerous characteristics to consider which, when all added together, will determine whether they match or not. One such characteristic is the name of a concept which is exploited by string-based approaches. The task of matching entity names has been explored by a number of communities, including statistics, databases, and artificial intelligence. A matching system uses several similarity measures which exploit different ontology characteristics in order to produce an alignment between ontologies. One of these characteristics are the names of the concepts in an ontology, which are exploited with syntactic similarities, more specifically string similarities, which are the focus of this paper. This paper will also make an extension to already existing techniques by taking into account the longest common substring when comparing two strings. All techniques discussed in this paper will be evaluated using the datasets by Cohen et al. [1] and the conference dataset originating from the 2010 Ontology Alignment Evaluation Initiative (OAEI) [4].

The rest of this paper is structured as follows. Section 2 will provide the reader with the necessary background information of this domain. Section 3 will detail the proposed extension of contemporary methods in this field. In section 4 the experiments performed with the results obtained will be presented. Section 5 will discuss the results obtained in chapter 4 and also propose future research. Section 6 will report the conclusions of the research performed.

## 2 Background information

### 2.1 Schemas and ontologies

The use of schemas originates from the field of databases, they are used to encode meta data, which is very useful to retrieve relevant data from a database. Later ontologies were developed which add more expressive ways to encode the meta data. Both methods are widely used in knowledge systems. There are some important differences and commonalities between schemas and ontologies as described by Shvaiko et al. [13], of which the keypoints are:

1. Database schemas often do not provide explicit semantics for their data. Semantics is usually specified explicitly at design-time, and frequently is not becoming a part of a database specification, therefore it is not available [11]. Ontologies are logical systems that themselves obey some formal semantics, e.g., we can interpret ontology definitions as a set of logical axioms.
2. Ontologies and schemas are similar in the sense that (i) they both provide a vocabulary of terms that describes a domain of interest and (ii) they both constrain the meaning of terms used in the vocabulary [6, 15].
3. Schemas and ontologies are found in such environments as the Semantic Web, and quite often in practice, it is the case that we need to match them.

Ontology mapping frameworks provide knowledge systems with the capacity to exchange information with other knowledge systems which use different ontologies. But before a framework can map ontologies, the system needs to ensure the interoperability of representations through transformations. There are several levels at which interoperability can be accounted for as described by Euzenat et al. [3].

1. Encoding: being able to segment the representation in characters.
2. Lexical: being able to segment the representation in words (or symbols).
3. Syntactic: being able to structure the representation in structured sentences (or formulas or assertions).
4. Semantic: being able to construct the propositional meaning of the representation.
5. Semiotic: being able to construct the pragmatic meaning of the representation (or its meaning in context).

### 2.2 Matching techniques categorization

Ontology mapping frameworks exploit multiple ontology characteristics during the matching process [13]. Matching techniques can compare two ontology concepts by utilizing information which describe the concepts themselves, or by investigating other related concepts, thus also exploiting the structure of an ontology. Techniques which utilize the structure of the ontology can be categorized as follows:

1. Graph-based techniques are graph algorithms which consider the input as labelled graphs. The considered ontologies are viewed as graph like structures containing terms and their inter-relationships. The comparison of a pair of nodes within the graph is usually based on their position within the graphs. The intuition behind is that, if two nodes are similar their adjacent nodes might also be similar.
2. Taxonomy-based techniques are also graph algorithms which consider only the specialization relation. The intuition behind this is that *is-a* links connect already similar terms, therefore their neighbouring nodes may also be somehow similar.
3. Repository of structures stores schemas/ontologies and their fragments together with pairwise similarities between them. When new structures are to be matched, they are first checked for similarity to the structures which are already available in the repository. The goal is to identify structures which are sufficiently similar to be worth matching in more detail, or reusing already existing alignments.
4. Model-based algorithms handle the input based on its semantic interpretation (e.g., model-theoretic semantics). Thus, they are well grounded deductive methods.

Matching techniques which do not use the structure of the ontologies can use different types of information about the concepts themselves. The techniques which use these different types of information can be divided into several categories:

1. String-based techniques are often used in order to match names and name descriptions of schema/ontology concepts. These techniques consider strings as sequences of letters in an alphabet. They are typically based on the following intuition: two concepts can be similar if their names are similar. Section 2.3 will go into further details about the string matching techniques.
2. Language-based techniques consider names as words in a natural language. They are based on Natural Language Processing techniques exploiting morphological properties of the input words.
3. Constraint-based techniques are algorithms which deal with the internal constraints being applied to the definitions of entities, such as types, cardinality of attributes, and keys.
4. Linguistic resources such as common knowledge or domain specific thesauri are used in order to match words based on linguistic relations between them (e.g., synonyms, hyponyms) [12]. In this case names of schema/ontology entities are considered as words of a natural language.
5. Alignment reuse techniques represent an alternative way of exploiting external resources, which contain in this case alignments of previously matched schemas/ontologies.
6. Upper level formal ontologies can be also used as external sources of common knowledge. The key characteristic of these ontologies is that they are logic-based systems, and therefore, matching techniques exploiting them can be based on the analysis of interpretations.

The listed techniques all have strengths and weaknesses with regard to the different heterogeneities which can exist between two ontology concepts. For instance a technique which uses linguistic resources can easily detect synonymous concepts but will be unable to handle concepts whose names contain spelling errors. Thus a combination of different techniques will be required to cope with all types of heterogeneities.

## 2.3 String similarities

The focus of this paper lies on the use of string similarities when applied to ontology mapping. Typically, these are applied to the names of concepts in order to produce a similarity matrix of correspondences. These can then be combined with similarity matrices stemming from difference measures, such that a alignment can be extracted.

String distance functions map a pair of strings  $s$  and  $t$  to a real number  $r$  where smaller values indicate a higher similarity between  $s$  and  $t$ . Similarity functions are analogues except that higher values of  $r$  indicate a higher similarity. To avoid confusion to the reader the value  $r$  is the one defined by similarity functions. The algorithms used to determine string similarities can be split up in multiple categories depending on their underlying logic to compare strings. First there are algorithms which look at the number of edit operations needed to transform one string into another for example the *Levenshtein* similarity [10]. Then there are algorithms which look at the number of matching characters in both strings for example the *Jaro* similarity [8]. Commonly, the *Winkler* extension [16], which increases the similarity of pairs of strings that have a common prefix, is applied to the *Jaro* similarity. Another category of algorithms are token based, strings are split up into tokens, like the *Jaccard* similarity [7]. There are also algorithms which combine multiple similarities to assign scores to pairs of strings, these are called hybrid similarity functions.

### 2.3.1 Levenshtein

One important subclass of distance functions are *Edit-distance functions*, which use the number of edit operations required to convert string  $s$  to string  $t$ . The most considered operations are character insertion, deletion, and substitution. Each of these operations will have a cost assigned to them. The costs assigned to an operation can be static or trained. We will consider the *Levenshtein* distance [10] which assigns a unit cost to each of the edit operations. Given strings  $s$  and  $t$ , the cost of an operation  $c_i$ , where  $i$  identifies the type of performed operation, and the quantity  $x_i$  which indicates how often an operation of type  $i$  needs to be

performed to convert  $s$  into  $t$ , the Levenshtein distance, which utilizes the three above mentioned operation, can be computed as follows:

$$Levenshtein(s, t) = \sum_{i=0}^3 c_i \cdot x_i \tag{1}$$

**2.3.2 Jaro**

The Jaro algorithm [8] is not based on edit operations but determines its similarity by looking at the number of matching characters between two strings and their relative position. Given two strings  $s = a_1, a_2...a_K$  and  $t = b_1, b_2...b_L$ , define a character  $a_i$  in  $s$  to be common with  $t$  when there is a  $b_j = a_i$  in  $t$  such that  $i - H \leq j \leq i + H$ , where  $H = \frac{\min\{|s|, |t|\}}{2}$ . Let  $s' = a'_1, a'_2...a'_{K'}$  be the characters in  $s$  which are common with  $t$  (in the same order they appear in  $s$ ) and let  $t' = b'_1, b'_2...b'_{L'}$  be analogous; now define a *transposition* for  $s', t'$  to be a position  $i$  such that  $a'_i \neq b'_i$ . Let  $T_{s', t'}$  be half the number of transpositions for  $s'$  and  $t'$ . The Jaro similarity is defined as:

$$Jaro(s, t) = \frac{1}{3} \cdot \left( \frac{|s'|}{|s|} + \frac{|t'|}{|t|} + \frac{|s'| - T_{s', t'}}{|s'|} \right) \tag{2}$$

**2.3.3 Jaro-Winkler**

A very well known extension to the Jaro algorithm is the Winkler extension [16]. This extension uses the length of the of the longest common prefix of  $s$  and  $t$  to assign more favourable ratings to pairs of strings which contain identical prefixes. This extension can be used in combination with any similarity but it is most commonly applied to the Jaro similarity. Let  $P$  be the length of longest common prefix, then define  $P' = \max(P, 4)$  then the Jaro-Winkler similarity is defined as:

$$Jaro-Winkler(s, t) = Jaro(s, t) + \frac{P'}{10} \cdot (1 - Jaro(s, t)) \tag{3}$$

**2.3.4 Jaccard**

This algorithm is a token-based distance measure, which can be applied to strings which have been preprocessed into tokens, called tokenization. Tokenization is the process of demarcating and possibly classifying sections of a string of input characters. The strings to be compared are considered to be multisets of words (or tokens). The *Jaccard similarity* [7] between two word sets  $S$  and  $T$  which is defined as:

$$Jaccard(S, T) = \frac{|S \cap T|}{|S \cup T|} \tag{4}$$

**2.3.5 SoftTFIDF**

Some background information is required in order to fully detail the *SoftTFIDF* similarity. The *TFIDF* [9] weighting scheme for document vectors, to which the *cosine* similarity is commonly applied, is a measure that is widely used in the information retrieval community for document retrieval. This measure depends, like the Jaccard similarity, on common elements between the two sets of tokens, but here the elements are weighted. The weights assigned to tokens  $w$  are larger when those tokens are rare in the collection of strings from which  $s$  and  $t$  are drawn. The similarity can then be defined as:

$$TFIDF(S, T) = \sum_{w \in S \cap T} V(w, S) \cdot V(w, T)$$

where  $V^l(w, S)$  is defined as the TF-IDF weight of the token  $w$  in the token vector of  $S$  and the function  $V(w, S) = V^l(w, S) / \sqrt{\sum_{w'} V^l(w', S)}$  is defined as the TF-IDF weight of token  $w$  related by the magnitude of the token vector of  $S$ . The SoftTFIDF algorithm, proposed by Cohen et al.[2], extends the notion of  $S \cap T$  such that it includes tokens which are similar according to a secondary similarity function. Since it utilizes a secondary similarity function denoted as  $sim'$  the SoftTFIDF can be categorized as a *hybrid similarity function*. Let  $CLOSE(\theta, S, T)$  be the set of words  $w \in S$  such that there is some  $v \in T$  such

that  $dist'(w, v) > \theta$ , and for  $w \in CLOSE(\theta, S, T)$  and let  $D(w, T) = \max_{v \in T} dist'(w, v)$ . Then the SoftTFIDF similarity is defined as:

$$SoftTFIDF(S, T) = \sum_{w \in CLOSE(\theta, S, T)} V(w, S) \cdot V(w, T) \cdot D(w, T) \quad (5)$$

### 3 Proposed extension

The proposed extension is mainly focused on ontology mapping but will also be benchmarked on other datasets containing real world data. This extension came to mind when studying the datasets in the field of ontologies, since concepts defined there are very likely to have high similarity because of the intuition when naming the concepts. To clarify this see the figure below, which is a small part of two ontologies in the OAEI dataset.

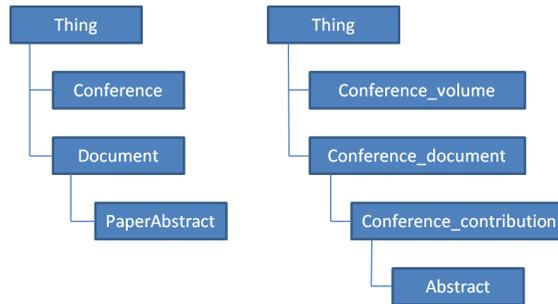


Figure 1: Two partial ontologies from the OAEI-conference dataset

These two example ontologies are part of the matching task using the OAEI dataset. A human inspecting these two example ontologies would quickly realize that the *Conference* and *Conference\_volume* denote the same meaning as well as the *Document* concept is the same as the *Conference\_document* concept. Likewise, the concepts *PaperAbstract* and *Abstract* also denote the same meaning. The Winkler extension to an algorithm only takes into account prefixes when comparing two strings, thus the corresponding concepts *PaperAbstract-Abstract* and *Document-Conference\_document* do not receive an increase of their similarity value when applying this extension. Intuitively, given that these pairs of names share a substring of considerable size, in this example at the suffix position, one would want allocate these pairs a higher similarity value opposed to their edit-distance based similarity. Hence, an extension is desired which also increases the similarity of strings if these share a non-prefix substring.

The extension researched in this paper utilizes the measure of the longest common substring between two strings, referred to as the LCS-Extension (Longest Common Substring). On its own this measure can be utilized as a string similarity as well, as evidenced by the reasearch of Stoilos et al. [14] performed on a benchmark dataset, making it a suitable candidate for combination with an edit distance. Whereas the Winkler extension is limited to the length of a common substring that is also a prefix of both strings, the proposed extension utilizes the length of the longest common substring regardless of its position in any of the two input strings. Let  $sim$  denote the similarity used as basis for the extension,  $LCS(s, t)$  be the length of the longest common substring of  $s$  and  $t$  and  $S$  a scaling factor such that  $0 \leq S \leq 1$ , the proposed LCS extension can then be defined as follows:

$$LCS-Extension(s, t) = sim(s, t) + \frac{LCS(s, t)}{\min(s, t)} \cdot S \cdot (1 - sim(s, t)) \quad (6)$$

The Winkler extension utilizes the length of the common prefix up to the length of 4 characters for the similarity adjustment. However, intuitively one could argue that the longer a common substring of two arbitrary strings is, the more likely it is that the meanings of these two strings correspond with each other. Hence, the proposed extension does not impose a limit on the computed substring length, but contrasts this length with the longest possible substring length, being the total length of the smaller of the two input strings. The proposed extension will be evaluated using different similarities as a basis.

## 4 Experiments

To compare the proposed extension with the other algorithms discussed in section 2.3, two datasets are used. The first dataset, stemming from the OAEI 2010 competition [4], contains a series of matching tasks between ontologies describing the conference domain, where the string metrics are applied to the names of the ontology concepts. The second dataset is a record-matching dataset, stemming from the research by Cohen et al. [2]. It contains a series of record matching tasks describing various domain, such as the names of animals and businesses.

### 4.1 Blocking method

When evaluating a similarity measure it is preferred to compute all pairwise similarities between two ontologies. This can result in large lists which are not computationally feasible. It is desired to pre-process the data, using so called blocking methods. For this research the same blocking method has been applied as in the evaluation by Cohen et al. [2] An example illustrating the intuition behind blocking: in statistical record linkage, it is common to group records by some variable which is known a priori to be usually the same for matching pairs. For example when matching records containing address information it is common to only consider pairs which have the same zip code.

The data used in this paper does not contain individuals for each concept, so there is little prior information available for pre-processing purposes. However the data is already partitioned into two mutually exclusive lists which reduces the number of pairs to be considered. To block this data, knowledge-free approaches are needed to reduce the number of considered pairs. The *blocking task* of two sets  $A$  and  $B$  selects all pairs of strings  $(s, t) \in A \times B$  such that  $s$  and  $t$  share some substring  $v$  which appears in at most a fraction  $f$  of all names. This method is called the token blocker. Another method for blocking the data is using n-grams to only consider strings which share an n-gram. For the moderate-size test sets considered here, we used  $f = 1$ . On the datasets which have been used in this research, the token blocker finds between 93.3% and 100.00% of the correct pairs for the different matching tasks, with an average of 98.9%.

### 4.2 Evaluation

The algorithms will all be evaluated using *precision* and *recall* values. These values are defined, in terms of true positives, false positives and false negatives of a retrieved list with regard to a reference list, as follows:

$$Precision = \frac{tp}{tp + fp} \quad (7) \qquad Recall = \frac{tp}{tp + fn} \quad (8)$$

Precision and recall are set-based measures, stemming from the field of information retrieval [5]. These evaluate the quality of an unordered set of retrieved ontology concepts according to their correctness and completeness. The investigated metrics will be evaluated using interpolated precision values at recall levels of 0.0, 0.1, ..., 0.9, 1.0, which are obtained by analysing the ranked list of retrieved correspondences. The rule to obtain the precision value at recall level  $i$  is to use the maximum precision obtained from the concept for any actual recall level greater than or equal to  $i$ . Note that the non-interpolated precision is not defined for recall values of 0, as opposed to the interpolated precision at recall level  $i = 0$ .

Before any of the similarities are evaluated on the datasets, these are blocked using the token blocker. All pairwise combinations of concepts are evaluated using the blocking method, after which the tested metrics are applied on the remaining pairs of concept names. The interpolated precision values for each mapping task are combined using the average interpolated precision.

### 4.3 Comparison with the Winkler extension

This experiment will compare the *Winkler* extension with the proposed *LCS* extension. This will show whether the intuition behind the proposed extension leads to a better performance than the more specific *Winkler* extension. Preliminary experiments revealed that a scaling factor of  $S = 0.8$  produced the highest performance for the *LCS* extension, with significant sub-par performances only observable at low values of  $S$  and  $S = 1$  The extensions were compared using both the *Jaro* and *Levenshtein* metric as base similarity.

The first comparison, seen in Figure 2, has been performed on the conference dataset. In the recall interval from 0 to 0.4 there is a minimal difference in the performance of the tested metrics, neither showing

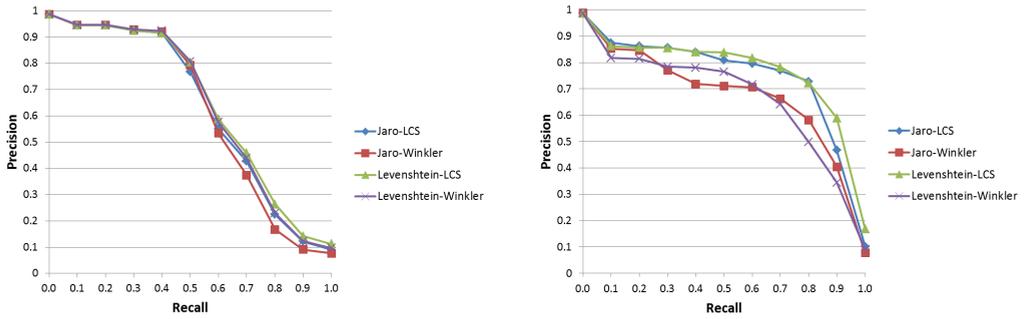


Figure 2: Comparison of the Winkler extension against the LCS extension when applied to two different base similarities on the Conference dataset (left) and Cohen dataset (right).

an advantage. The recall interval of 0.4 and 0.6 displays more pronounced differences, of which the most notable is that the *Jaro-LCS* metric performed slightly worse than the remaining metrics.

From a recall values of 0.6 and higher it appears that the proposed extension displays a superior performance with regard to the Winkler extension applied to the same base similarity.

When comparing the metrics on the Cohen data set, see Figure 2, the proposed *LCS* extension outperforms both *Winkler extension* based metrics by a substantial percentage. The *Levenshtein-Winkler* metric performs worse at a recall of 0.1 whereas the *Jaro-Winkler* performs almost similar up until a recall of 0.2. At the remaining recall values the *LCS extension* outperforms the *Winkler extension* by a significant margin, peaking at recall values of 0.8 and 0.9 with an increase of precision of at least 0.1.

#### 4.4 Comparison with other measures

In this experiment, the best performing configuration of our proposed extension is compared to other established methods from this field. The *LCS extension* will be applied to the *Levenshtein* similarity, due to its superior performance as seen in sub-section 4.3.

The performed evaluation on the conference data set, see figure 3, reveals that the token based *Jaccard similarity* displays the worst performance of the tested metrics. The hybrid *SoftTFIDF* metric is performing slightly worse than the edit based distances on lower recall values, but displays a superior performance on higher recall values. The edit-based distances all display a similar performance curve, with some of them performing strictly better, of which the *Levenshtein-LCS* performs best considering all the recall values.

The comparison shown in figure 3 is obtained by comparing all the algorithms on the Cohen dataset. It is evident that on this data set token-based distance functions outperform the majority of the edit-based

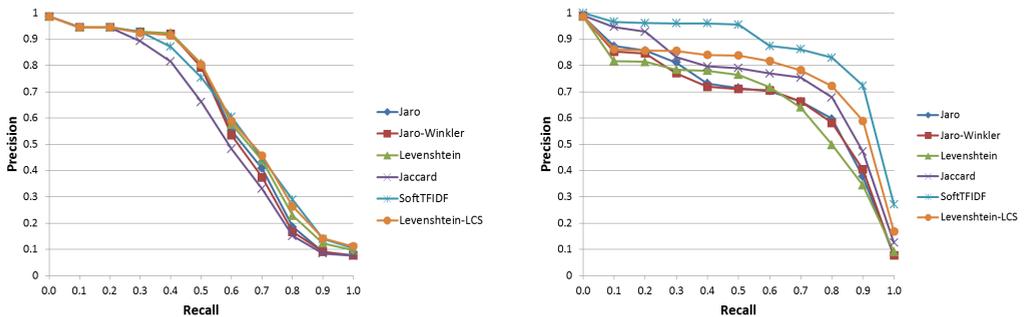


Figure 3: Comparison of all tested similarity measures on the Conference dataset (left) and Cohen dataset (right)

distance functions, especially at lower recall values. Since the *SoftTFIDF* combines a token-based approach with an edit-distance based approach, by using an edit-distance metric as secondary distance function, it outperforms all tested metrics by a significant margin. The *Jaccard similarity* outperforms all tested edit-distance based metrics for recall values up to 0.3. However, for recall values of 0.3 and higher, the proposed extension applied to the *Levenshtein* metric significantly outperforms the tested edit-distance based metrics as well as the *Jaccard* metric.

## 5 Conclusion

In this paper, we proposed a generalization of the Winkler extension using the measure of the longest common sub-string. We used the Jaro and Levenshtein similarity as base in order to compare our generalization with the Winkler extension. The experiments show that our extension outperforms the Winkler extension for either base similarity on both datasets, the differences being more pronounced when evaluating the record-matching dataset. Contrasting the proposed extension with contemporary metrics revealed that it outperforms all tested metrics, except for the hybrid *SoftTFIDF* metric.

The proposed extension has been applied to edit-distance based functions in the performed experiments. Future research could investigate the potential improvements of the extension when being incorporated into a hybrid distance function. Also, it is possible that a performance gain can be achieved by analysing the input strings for stop-words, such that concept names for which the common substring is a stop word do not receive an increased similarity value.

## References

- [1] W. W. Cohen. Data integration using similarity joins and a word-based information representation language. *ACM Trans. Inf. Syst.*, 18(3):288–321, July 2000.
- [2] W. W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string distance metrics for name-matching tasks. 2003.
- [3] J. Euzenat. Towards a principled approach to semantic interoperability. In Asuncin Gmez Prez, Michael Gruninger, Heiner Stuckenschmidt, and Michael Uschold, editors, *Proc. IJCAI 2001 workshop on ontology and information sharing, Seattle (WA US)*, pages 19–25, 2001.
- [4] J. Euzenat, A. Ferrara, C. Meilicke, A. Nikolov, J. Pane, F. Scharffe, P. Shvaiko, H. Stuckenschmidt, O. Svab-Zamazal, V. Svatek, and C. Trojahn. Results of the ontology alignment evaluation initiative 2010. 2010.
- [5] F. Giunchiglia, M. Yatskevich, P. Avesani, and P. Shvaiko. A large dataset for the evaluation of ontology matching. *Knowl. Eng. Rev.*, 24:137–157, June 2009.
- [6] N. Guarino. The role of ontologies for the semantic web (and beyond). Technical report, Institute for Cognitive Sciences and Technology (ISTCCNR), 2004.
- [7] P. Jaccard. Nouvelles recherches sur la distribution florale. *Bulletin de la Société Vaudaise des Sciences Naturelles*, 44:223–270, 1908.
- [8] M. A. Jaro. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.
- [9] K. Sparck Jones. Document retrieval systems. chapter A statistical interpretation of term specificity and its application in retrieval, pages 132–142. Taylor Graham Publishing, London, UK, 1988.
- [10] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8, 1966.
- [11] N. F. Noy and M. Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems*, 6:428–440, 2004.
- [12] F. C. Schadd and N. Roos. Improving ontology matchers utilizing linguistic ontologies: an information retrieval approach. In *Proceedings of the 23rd Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2011)*, 2011.
- [13] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. In Stefano Spaccapietra, editor, *Journal on Data Semantics IV*, volume 3730 of *Lecture Notes in Computer Science*, pages 146–171. Springer Berlin / Heidelberg, 2005.
- [14] G. Stoilos, G. Stamou, and S. Kollias. A string metric for ontology alignment. In *Proceedings of ISWC*, pages 624–637, 2005.
- [15] M. Uschold and M. Gruninger. Ontologies and semantics for seamless connectivity. *SIGMOD Rec.*, 33(4):58–64, December 2004.
- [16] W. E. Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Technical report, 1990.

# Structure Approximation of Most Probable Explanations in Bayesian Networks

Johan Kwisthout

*Leiden University, Leiden Institute for Advanced Computing Science,  
P.O. Box 9512, 2300RA Leiden, The Netherlands, johank@liacs.nl*

## Abstract

Typically, when one discusses approximation algorithms for (NP-hard) problems (like TRAVELING SALES-PERSON, VERTEX COVER, KNAPSACK), one refers to algorithms that return a solution whose *value* is (at least ideally) close to optimal; e.g., a tour with almost minimal length, a vertex cover of size just above minimal, or collection of objects that has close to maximal value. In contrast, one might also be interested in approximations algorithms that return solutions that *resemble* the optimal solutions, i.e., whose *structure* is akin to the optimal solution, like a tour that is almost similar to the optimal tour, a vertex cover that differs in only a few vertices from the optimal cover, or a collection that is similar to the optimal collection. In this paper, we discuss structure-approximation of the problem of finding the most probable explanation of observations in Bayesian networks, i.e., finding a joint value assignment that *looks like* the most probable one, rather than has an *almost as high value*. We show that it is NP-hard to obtain the value of *just a single variable* of the most probable explanation. However, when partial orders on the values of the variables are available, we can improve on these results.

## 1 Introduction

A key computational problem in Bayesian networks [17] is the computation of the *most probable explanation* (MPE) of a set of observed phenomena; i.e., given a Bayesian network whose variables are partitioned into an *evidence* set  $\mathbf{E}$  with observed joint value assignment  $\mathbf{e}$  and an *explanation* set  $\mathbf{M}$ , determine the joint value assignment  $\mathbf{m}$  to the explanation set  $\mathbf{M}$  such that  $\Pr(\mathbf{M} = \mathbf{m}, \mathbf{E} = \mathbf{e})$  is maximal. This problem, also called Bayesian abduction, is a key component in many decision support systems like [15, 21], in many Bayesian models of cognition, for example intention recognition [2] or recipient design [22], as well as in various models of sociological [19] or economical [8] processes.

Unfortunately, computing the MPE is in general NP-hard [13, 3, 18] and remains NP-hard when the most probable explanation is to be *approximated* rather than exactly computed. In particular it is NP-hard to find a joint value assignment whose probability is within a fixed ratio of the most probable joint value assignment [1] and it is even NP-hard to find a joint value assignment that has a non-zero probability [13]. However, these formal notions of approximation focus on the *value* of the explanation, i.e., the goal is to find an explanation whose *probability* is ‘close’ to the probability of the most probable explanation. Sometimes we may not be primarily interested in finding explanations with an almost-as-high probability, but rather in explanations that are *quite similar* to the most probable explanation, that is, they *look like* the most probable explanation. For example, in cognitive science, one’s goal is to describe, model, and predict human cognition. In such applications it is conceivable that we are most interested in approximating structure, rather than value [16]; we will refer to this notion of approximation as *structure approximation*.

Preferably, of course, in many domains we would like to have an approximation that both resembles the optimal solution *and* have an almost-as-high probability [4]. While it may well be the case that ‘good’ value approximations sometimes have a similar structure as the optimal solution, this need not be the case, as we will show in Section 2.3.

Structure approximation has its roots in computational complexity theory<sup>1</sup>[12, 6]. The relevance of

---

<sup>1</sup>Here it was called *witness* approximation, referring to the more general concept of a witness or certificate: a string that can be used to verify membership in NP. Such a string may (but does not need to) encode an actual solution, such as a satisfying truth instantiation.

structure approximation, in particular in the context of the so-called Coherence Problem, was first suggested by Millgram [16] and extensively studied in Hamilton et al. [9] and Van Rooij et al. [23]. In this paper we further build on this work and discuss structure approximations of MPE<sup>2</sup>.

In the remainder of this paper, we will discuss some relevant preliminaries and definitions in Bayesian networks and structure approximation in Section 2. In Section 3 we focus on structure-approximating MPE. We discuss the computational complexity of structure approximation of MPE in general in Subsection 3.1, and the effect of having an *ordering* of the variables in Subsection 3.2. In Section 4 we conclude this paper.

## 2 Preliminaries

In this section we introduce Bayesian networks and, more in particular, the problem of finding the most probable explanation (MPE) for a subset of variables in the network, given observations for the other variables. For more background, the reader is referred to textbooks as [17, 10, 11] and overview papers as [14, 13]. Furthermore, we introduce a formal definition of structure approximation, as presented in [9]. We assume that the reader is familiar with basic notions in complexity theory, like NP-hardness proofs; for more background, we refer to [7].

### 2.1 Bayesian networks and the MPE problem

A Bayesian or probabilistic network  $\mathcal{B}$  is a graphical structure that models a set of stochastic variables, the conditional independencies among these variables, and a joint probability distribution over these variables.  $\mathcal{B}$  includes a directed acyclic graph  $\mathbf{G}_{\mathcal{B}} = (\mathbf{V}, \mathbf{A})$ , modeling the variables and conditional independencies in the network, and a set of parameter probabilities  $\Gamma$  in the form of conditional probability tables (CPTs), capturing the strengths of the relationships between the variables. The network models a joint probability distribution  $\Pr(\mathbf{V}) = \prod_{i=1}^n \Pr(V_i \mid \pi(V_i))$  over its variables, where  $\pi(V_i)$  denotes the parents of  $V_i$  in  $\mathbf{G}_{\mathcal{B}}$ . We will use upper case letters to denote individual nodes in the network, upper case bold letters to denote sets of nodes, lower case letters to denote value assignments to nodes, and lower case bold letters to denote joint value assignments to sets of nodes. We will use  $\mathbf{E}$  to denote a set of evidence nodes, i.e., a set of nodes for which a particular joint value assignment  $\mathbf{e}$  is observed; likewise, we will use  $\mathbf{M}$  to denote a set of nodes for which the explanation is sought. We will sometimes write  $\Pr(\mathbf{x})$  as a shorthand for  $\Pr(\mathbf{X} = \mathbf{x})$  if no ambiguity can occur. We denote with  $\Omega(X)$  the set of all values that  $X$  can take;  $\Omega(\mathbf{X})$  is defined analogously for sets of variables.

Among other computational problems defined on Bayesian networks, one particularly interesting problem for many applications is the problem of determining the *most probable explanation* for some observations, i.e., the most probable joint value assignment to a subset of variables in the network, given evidence for the other variables<sup>3</sup>. This problem is formally defined as follows [13].

MPE

**Instance:** A probabilistic network  $\mathcal{B} = (\mathbf{G}_{\mathcal{B}}, \Gamma)$ , where  $\mathbf{V}$  is partitioned into a set of evidence nodes  $\mathbf{E}$  with a joint value assignment  $\mathbf{e}$ , and an explanation set  $\mathbf{M}$ .

**Output:**  $\operatorname{argmax}_{\mathbf{m}} \Pr(\mathbf{m}, \mathbf{e})$ , i.e., the most probable joint value assignment  $\mathbf{m}$  to the nodes in  $\mathbf{M}$  and evidence  $\mathbf{e}$ , or the designated symbol  $\perp$  if  $\Pr(\mathbf{m}, \mathbf{e}) = 0$  for every joint value assignment  $\mathbf{m}$  to  $\mathbf{M}$ .

MPE is intractable in general; to be precise, the problem is  $\text{FP}^{\text{NP}}$ -complete and has an NP-complete decision variant [13, 18].

### 2.2 Structure approximation

The notion of a structure approximation is typically captured using a *solution distance function*, a metric associated with each optimization problem relating candidate solutions with the optimal solution [9]. Let  $\Pi$  be a optimization problem with instance  $x$ , let  $\operatorname{cansol}(x)$  denote a function returning candidate solutions to

<sup>2</sup>Note that the term ‘structure’ does *not* refer to the graphical structure (i.e., the arcs) of the network, but to the structure of the joint value assignments.

<sup>3</sup>If we have only partial evidence, i.e., the network is partitioned into variables for which the explanation is sought, evidence variables, and other variables that constitute neither evidence nor explanation, then the problem generalized to a Partial (or Marginal) MAP problem. The (intractability) results presented here generalize also to Partial MAP.

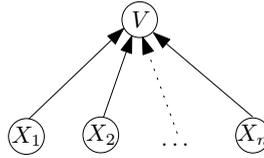


Figure 1: Example network with distinct structure and value approximations

$x$ , with  $optsol(x)$  denoting a function returning the *optimal* solution<sup>4</sup> to  $x$ . For any  $y, y' \in cansol(x)$ , let  $d(y, y')$  be the distance between  $y$  and  $y'$  as defined by  $d$ . As  $d$  is a metric, the following properties hold for all  $a, b, c \in cansol(x)$ :

1.  $d(a, a) = 0$
2. if  $a \neq b, d(a, b) > 0$
3.  $d(a, b) = d(b, a)$
4.  $d(a, b) + d(b, c) \geq d(a, c)$

Typically, for many problems  $\Pi$ ,  $d$  might correspond to the *Hamming distance* or *edit distance* between two candidate solutions: the number of elements in which the candidate solutions differ, or the number of operations needed to transform one candidate solution into another. We define a  $h/d$ -structure approximation of  $\Pi$  as follows:

**Definition 2.1 ([9]).** *Given an optimization problem  $\Pi$ , a solution-distance function  $d$ , and a non-decreasing function  $h : \mathbb{N} \rightarrow \mathbb{N}$ , an algorithm  $A$  is a polynomial-time  $h/d$ -structure approximation algorithm if for every instance  $x$  of  $\Pi$ ,  $d(A(x), optsol(x)) \leq h(|x|)$  and  $A$  runs in time polynomial in  $|x|$ .*

### 2.3 Value versus structure approximation

Possibly counter to intuition, a “good” value approximation may not be a “good” structure approximation and vice versa. As an example, consider the Bayesian network in Figure 1 with binary variables  $V, X_1, \dots, X_n$ , a uniform probability distribution for the variables  $X_1$  to  $X_n$ , and the following conditional probability distribution for  $V$ :

$$\Pr(V = \text{TRUE}, X_1, \dots, X_n) = \begin{cases} 1 & \text{if } \forall_i X_i = \text{TRUE} \\ 1 - \epsilon & \text{if } \forall_i X_i = \text{FALSE} \\ 0 & \text{otherwise} \end{cases}$$

Note that the most probable explanation for the observation  $V = \text{TRUE}$  would be the explanation where all variables  $X_i$  are set to **TRUE**, and the second most probable explanation where all variables  $X_i$  are set to **FALSE**. Any non-zero value approximation thus would yield an explanation with a completely different structure than the most probable explanation. On the other hand, any explanation that has a similar structure (i.e., differ in only few variables) would have a probability of zero.

## 3 Structure approximation of MPE

Let  $cansol(\mathcal{B}, \mathbf{e})$  denote the set of explanations (i.e., joint value assignments to  $\mathbf{M}$ ) of a Bayesian network  $\mathcal{B}$  with observed evidence  $\mathbf{e}$ , with  $optsol(\mathcal{B}, \mathbf{e})$  as the most probable explanation, i.e., the joint value assignment to  $\mathbf{M}$  with the highest joint probability. We define the structure distance function  $d_H(\mathbf{m}, optsol(\mathcal{B}, \mathbf{e}))$  as the Hamming distance between explanation  $\mathbf{m} \in cansol(\mathcal{B}, \mathbf{e})$  and the most probable explanation.

In the remainder of this paper, we consider  $h$  to be a function taking an MPE instance  $x = \{\mathcal{B}, \mathbf{e}\}$  and returning a distance. With  $h(x)/d_H$ -structure-approximate-MPE, we define the problem of finding a structure approximation that differs in *at most*  $h(x)$  variables from the most probable explanation  $optsol(\mathcal{B}, \mathbf{e})$ . With  $E(h(x))/d_H$ -structure-approximate-MPE we define the problem of finding a joint value assignment

<sup>4</sup>Or, in case of a draw, one of the optimal solutions.

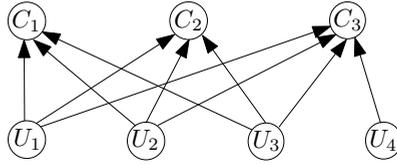


Figure 2: Construction of  $\mathcal{B}_{\phi_{\text{ex}}}$  from  $\phi_{\text{ex}}$

that has an *expected* Hamming distance  $h(x)$  to  $\text{optsol}(\mathcal{B}, e)$ , i.e., a structure approximation is sought that differs *on average* in at most  $h(x)$  variables from the MPE.

### 3.1 Computational complexity

In this section we will discuss the computational complexity of structure approximations of MPE. Note that a *random guess* of the values of variables would return a value assignment which gives an *expected* Hamming distance  $h(x) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c}$ , with  $c$  as the cardinality of the (unobserved) variables. In particular, when all unobserved variables are binary, we can expect to guess half of them correctly.

**Corollary 3.1.** *MPE is  $E(h(x))/d_H$ -structure approximable for  $h(x) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c}$ .*

We cannot expect to do better than chance: given that it is NP-hard to  $\frac{n}{2} - \epsilon/d_H$ -structure approximate 3SAT [6] and we can reduce 3SAT to MPE in polynomial time while preserving the structure of the certificates (by a simple variant of the proof used in [13, p.1457], which is omitted here for reasons of space), any polynomial-time  $|\mathbf{M}| - \frac{|\mathbf{M}|}{c} - \epsilon/d_H$ -structure approximation algorithm for MPE could be used to find a  $\frac{n}{2} - \epsilon/d_H$ -structure approximation of any 3SAT instance in polynomial time.

**Lemma 3.2.** *MPE is  $h(x)/d_H$ -structure inapproximable for  $h(x) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c} - \epsilon$ , unless  $P = NP$ .*

This result holds for binary variables with indegree at most three<sup>5</sup>. Here, we allow the approximation algorithm to select the  $h(x)$  variables. If we are allowed to *designate* the variables for which the value is sought, then it is easy to see that we cannot have a polynomial-time structure approximation algorithm  $A$  for MPE, even for a single variable, unless  $P = NP$ , as we could use  $A$  consecutively for all  $|\mathbf{M}|$  unobserved variables of  $\mathcal{B}$  and thus obtain a polynomial-time exact algorithm for MPE; as MPE is NP-hard, the result follows as a corollary. However, we can prove a much stronger result for networks with three values per variable and indegree at most six: There cannot exist an algorithm that tells<sup>6</sup> us the value of an *arbitrary* single variable, unless  $P = NP$ :

**Theorem 3.3.** *No algorithm can calculate the value of one of the variables in the most probable explanation in polynomial time, unless  $P = NP$ .*

We will prove Theorem 3.3 with a reduction from 3SAT, defined as follows.

#### 3-CNF SATISFIABILITY (3SAT)

**Instance:** A Boolean formula  $\phi = (U, C)$  in 3-CNF form, with variables  $U = \{u_1, \dots, u_n\}$  and literals  $C = \{c_1, \dots, c_m\}$ .

**Question:** Does there exist a truth assignment to the variables  $U$  such that all clauses  $C$  are satisfied?

As a running example, we will construct a network for the following (satisfiable) 3SAT instance [5]:

**Example 3.4.**  $\phi_{\text{ex}} = (U, C)$ , where  $U = \{u_1, u_2, u_3, u_4\}$ , and  $C = \{(u_1 \vee u_2 \vee u_3), (\neg u_1 \vee \neg u_2 \vee u_3), (u_2 \vee \neg u_3 \vee u_4)\}$ .

<sup>5</sup>As each clause has three variables, the corresponding MPE instance has indegree at most three.  
<sup>6</sup>Note that here we require that the algorithm not only returns a joint value assignment  $\text{cansol}(x)$ , but also tells us *which subset* of  $\text{cansol}(x)$  matches  $\text{optsol}(x)$ .

We construct a Bayesian network  $\mathcal{B}_\phi$  from a 3SAT instance  $\phi = (U, C)$  as follows. For each variable  $u_i$  in  $\phi$  we add a *ternary* stochastic variable  $U_i$  in  $\mathcal{B}_\phi$  with values  $\{\text{TRUE}, \text{FALSE}, \#\}$  and uniform prior probability; the set of all  $U_i$  is denoted  $\mathbf{U}$ . For each clause  $c_j$  in  $\phi$  we add a binary stochastic variable  $C_j$  in  $\mathcal{B}_\phi$  with values  $\text{TRUE}$  and  $\text{FALSE}$ ; the set of all  $C_j$  is denoted  $\mathbf{C}$ .  $C_j$  is to be conditioned on the variables  $\mathbf{U}_j = \{U_j^1, U_j^2, U_j^3\}$  that correspond to the variables that occur in  $c_j$ , and (for  $j > 1$ ) on the variables  $\mathbf{U}_{j-1} = \{U_{j-1}^1, U_{j-1}^2, U_{j-1}^3\}$  that correspond to the variables that occur in  $c_{j-1}$ . To improve readability, we define the following shorthands for joint value assignments to  $\mathbf{U}_j$  and  $\mathbf{U}_{j-1}$ : let  $\mathbf{u}_\#$  denote a joint value assignment where *all* variables have the value  $\#$ , and let  $\mathbf{u}_{\text{TF}}$  denote a joint value assignment where *none* of the variables have the value  $\#$ , i.e., all are  $\text{TRUE}$  or  $\text{FALSE}$ . For  $C_j (j > 1)$  the following conditional probability distribution is defined.

$$\Pr(C_j = \text{TRUE} \mid \mathbf{U}_j, \mathbf{U}_{j-1}) = \begin{cases} 1 & \text{if } \mathbf{U}_j = \mathbf{u}, \text{ where } \mathbf{u} \text{ makes clause } C_j \text{ true, and } \mathbf{U}_{j-1} = \mathbf{u}_{\text{TF}} \\ \epsilon & \text{if } \mathbf{U}_j = \mathbf{u}_\# \text{ and } \mathbf{U}_{j-1} = \mathbf{u}_\# \\ 0 & \text{otherwise} \end{cases}$$

Here,  $\epsilon$  is defined to be a sufficiently small (i.e.,  $\epsilon < \frac{1}{2^n}$ ), yet polynomial-time computable, value. Likewise,  $C_1$  is defined as follows.

$$\Pr(C_1 = \text{TRUE} \mid \mathbf{U}_1) = \begin{cases} 1 & \text{if } \mathbf{U}_1 = \mathbf{u}, \text{ where } \mathbf{u} \text{ makes clause } C_1 \text{ true} \\ \epsilon & \text{if } \mathbf{U}_1 = \mathbf{u}_\# \\ 0 & \text{otherwise} \end{cases}$$

As an example of this construction, Figure 2 shows the network as constructed from  $\phi_{\text{ex}}$ . We set the evidence variables  $\mathbf{E} = \mathbf{C}$  with  $\mathbf{e} = \bigwedge_{j=1}^m C_j = \text{TRUE}$ . We claim that  $\phi$  is satisfiable if and only if *none* of the variables in the most probable joint value assignment  $\mathbf{u}$  to  $\mathbf{U}$  has the value  $\#$ , and unsatisfiable if and only if *all* of the variables in  $\mathbf{u}$  have the value  $\#$ . Thus, if an approximation algorithm tells us the value of *any* variable of the most probable explanation of  $\mathcal{B}$ , we can use that algorithm to solve the corresponding 3SAT instance in polynomial-time.

*Proof of Theorem 3.3.* Assume there exists a polynomial-time structure approximation algorithm  $A$  that, when given an MPE instance, returns for one of the variables in the explanation set  $M$  a value that corresponds to the value of that variable in the most probable explanation. We will show that  $A$  can be used to decide 3SAT in polynomial time; hence, from the existence of such an algorithm it would follow that  $P = NP$ . Let  $\phi$  be an arbitrary instance of 3SAT and let  $(\mathcal{B}_\phi, \mathbf{E}, \mathbf{e})$  be the MPE instance as constructed above. Note that we can construct  $\mathcal{B}_\phi$  from  $\phi$  in polynomial time, as every literal and clause in  $\phi$  corresponds to a single variable in  $\mathcal{B}_\phi$  and the size of the conditional probability tables of each variable is bounded by a constant.

Let  $\mathbf{u}$  be a joint value assignment to the variables of  $\mathbf{U}$  of  $\mathcal{B}_\phi$ . We will distinguish between three possible scenarios:

1.  $\mathbf{u} \in \{\#\}^n$ , i.e., *all* variables are set to  $\#$
2.  $\mathbf{u} \in \{\text{TRUE}, \text{FALSE}\}^n$ , i.e., *none* of the variables are set to  $\#$
3.  $\mathbf{u} \in \{\text{TRUE}, \text{FALSE}, \#\}^n$

Note that in case 3)  $\Pr(\mathbf{u}, \mathbf{e}) = 0$  due to the constraints in the joint probability distributions of  $C_j$ . In case 2), if  $\mathbf{u}$  does *not* satisfy  $\phi$ , then also  $\Pr(\mathbf{u}, \mathbf{e}) = 0$ . If on the other hand  $\mathbf{u}$  *does* satisfy  $\phi$ , then the probability  $\Pr(\mathbf{u}, \mathbf{e})$  equals  $\frac{1}{N_{\text{sat}}(1+\epsilon)}$ , where  $1 \leq N_{\text{sat}} \leq 2^n$  denotes the number of satisfying truth assignments to  $\phi$ . In case 1), if  $\phi$  is satisfiable, then  $\Pr(\mathbf{u}, \mathbf{e}) = \frac{\epsilon}{1+\epsilon}$ ; as  $\epsilon$  was chosen to be strictly less than  $\frac{1}{2^n}$ , this probability is lower than the probability of any satisfying joint value assignment. However, when  $\phi$  is not satisfiable, then  $\Pr(\mathbf{u}, \mathbf{e}) = 1$ .

Thus, the most probable explanation for evidence  $\mathbf{e} = \bigwedge_{j=1}^m C_j = \text{TRUE}$  is either  $\mathbf{u} \in \{\text{TRUE}, \text{FALSE}\}^n$  if  $\phi$  is satisfiable, or  $\mathbf{u} \in \{\#\}^n$  if  $\phi$  is not satisfiable. Now assume that, when given  $(\mathcal{B}_\phi, \mathbf{E}, \mathbf{e})$  as input,  $A$  outputs the value assignment of one of the unobserved variables in  $\mathcal{B}_\phi$ , that correspond to the value in the most probable explanation of  $\mathcal{B}_\phi$ . In case  $A$  outputs  $\text{TRUE}$  or  $\text{FALSE}$ ,  $\phi$  is satisfiable; in case  $A$  outputs  $\#$ ,  $\phi$  is not satisfiable. Hence, we can use  $A$  to solve 3SAT in polynomial time, concluding the proof.  $\square$

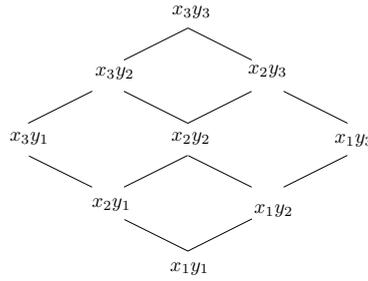


Figure 3: A lattice describing the partial order of the joint value assignments to the variables  $X$  and  $Y$

### 3.2 Ordered variables

We saw in the previous section that it is NP-hard to structure-approximate even a single variable of the most probable explanation in a Bayesian network. However, we assumed that the values of the variables in the network were *unordered*. In this section we assume a particular order on the values and investigate the consequences for the computational complexity of structure approximation.

Typically, in a Bayesian network some variables might have a ‘natural’ ordering, like a variable HEIGHT with values TALL, NORMAL and SMALL; these values are ordered  $\text{SMALL} \preceq \text{NORMAL} \preceq \text{TALL}$ . Other variables, like BLOODTYPE or ETHNICGROUP lack such an ordering. When a variable is ordered, it makes sense to redefine the distance measure: when HEIGHT is assigned the value TALL in the most probable explanation, NORMAL would be a better approximation than SMALL.

In the remainder we assume that all variables are ordered, and we introduce a *partial ordered lattice* [20] and a corresponding *lattice distance function*. The lattice includes all joint value assignments to the observable variables in the network and it captures the partial order between the assignments. The bottom of the lattice encodes the joint value assignment  $\mathbf{m}$  such that  $\mathbf{m} \preceq \mathbf{m}'$  for all  $\mathbf{m}' \in \Omega(\mathbf{M})$ . Likewise, the top of the lattice encodes the joint value assignment  $\mathbf{m}''$  such that  $\mathbf{m}' \preceq \mathbf{m}''$  for all  $\mathbf{m}' \in \Omega(\mathbf{M})$ . In general, a lattice element  $L(\mathbf{m})$  encoding a joint value assignment  $\mathbf{m}$  precedes another lattice element  $L(\mathbf{m}')$  if and only if  $\mathbf{m} \preceq \mathbf{m}'$ . In Figure 3 an example (from [20]) is shown for two ternary variables  $X$  and  $Y$ .

A natural distance function comparing two joint value assignments  $\mathbf{m}$  and  $\mathbf{m}'$  would be the *distance in the lattice* between these assignments, i.e., the length of the shortest path from  $L(\mathbf{m})$  to  $L(\mathbf{m}')$ . For example, the distance between  $x_2y_1$  and  $x_1y_3$  would be three. Note that this distance function, denoted by  $d_L$ , is a metric as the properties of Section 2.2 also hold for  $d_L$ . Using this distance function, we can find a trivial *guaranteed*  $h(x)/d_L$ -structure approximation with ordering for  $h(x) = |\mathbf{M}| \cdot \lceil \frac{2}{c} \rceil$ , rather than the *expected*  $E(h(x)) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c}$  without ordering, by always picking the ‘middle’ value in the order. We can, however, not expect to do better than  $h(x) = |\mathbf{M}|$  for  $c \geq 5$ , unless  $\text{P} = \text{NP}$ :

**Theorem 3.5.** *MPE is  $h(x)/d_L$ -structure inapproximable for  $h(x) = |\mathbf{M}| - 1$ , unless  $\text{P} = \text{NP}$ .*

*Proof.* Similar as in the proof of Theorem 3.3, and using the same construction, we show that the existence of a polynomial-time algorithm  $A$  that can  $h(x)/d_L$ -structure-approximate MPE for  $h(x) = |\mathbf{M}| - 1$  implies that we can decide 3SAT in polynomial time. We augment the construction used to prove Theorem 3.3 as follows: let all variables  $U_i$  have five values  $\Omega(U_i) = \{\text{FALSE}, \text{TRUE}, \#, d_1, d_2\}$  in which  $d_1$  and  $d_2$  act as dummy variables.  $U_i$  is uniformly distributed, and the order of  $\Omega(U_i)$  is  $\text{FALSE} \preceq d_1 \preceq \# \preceq d_2 \preceq \text{TRUE}$ . The conditional probability distribution of  $C_j$  is unaltered. We claim that, for any  $h(x)/d_L$ -structure approximation with  $h(x) \leq |\mathbf{M}| - 1$ , the majority of the variables that contain non-dummy values can be used to decide satisfiability of  $\phi$ : if the (strict) majority of these variables has TRUE or FALSE as value, then the instance is satisfiable, otherwise the instance is unsatisfiable.

Observe that an approximation with  $h(x) = |\mathbf{M}| - 1$  has at least *one* ‘correct’ variable, as any deviation from the MPE would increase  $h(x)$  by at least one, i.e., every variable that has a value that is not equal to the MPE contributes a distance of 1 to  $h(x)$ . In particular, when one of the variables is correctly labeled with either  $\#$  (for an unsatisfying instance) or TRUE or FALSE (for a satisfying instance), and the other variables have dummy values that are closest to the MPE value of that variable (i.e.,  $d_1$  for FALSE,  $d_2$  for TRUE, and

either  $d_1$  or  $d_2$  for  $\#$ ), then  $h(x) = |\mathbf{M}| - 1$ ; clearly here a majority of the (non-dummy) variables correctly reflects the satisfiability of the instance.

Now we show that this property holds for every alteration to this joint value assignment that maintains that  $h(x) = |\mathbf{M}| - 1$ . We will demonstrate the case that  $\phi$  is satisfiable; for unsatisfiable  $\phi$ , the proof goes analogously.

- If we replace a dummy value with a  $\#$  value, then  $h(x)$  increases by one. We must also change another dummy value to TRUE or FALSE (whichever is closest) to maintain that  $h(x) = |\mathbf{M}| - 1$ , so still the majority of non-dummy variables has as value TRUE or FALSE.
- If we replace a TRUE or FALSE value to a  $\#$  value, then  $h(x)$  increases by two, and so two dummy variables need to be changed into TRUE or FALSE.

Thus, if  $A$  returns a  $h(x)/d_L$ -structure approximation with  $h(x) \leq |\mathbf{M}| - 1$ , then we can use the output to decide 3SAT: count the number TRUE or FALSE values and the number of  $\#$ -values. If the first number is higher than the second, answer *yes*, else answer *no*. As  $A$  runs in polynomial time, this algorithm can decide 3SAT in polynomial time, hence  $P = NP$ .  $\square$

## 4 Conclusion

In this paper we discussed structure approximations of MPE. In general, we cannot do better than just randomly guess the joint value assignment: we then would on average expect to guess  $\frac{1}{c}$  of the variables correctly, where  $c$  is the cardinality of the variables. As it is NP-hard to determine the value of more than  $\frac{1}{c}$  of the variables in the MPE, there is little room for improvement. We hypothesize (but could not prove) that it is even NP-hard to get an *expected* structure approximation that is strictly better than  $|\mathbf{M}| - \frac{|\mathbf{M}|}{c}$ .

Furthermore, we showed that it is NP-hard in general to obtain an approximation that determines even a *single* variable in the MPE. So, without information on the ordering of the values or restrictions on the network structure or probability distribution, if we want information on the structure of the MPE (in polynomial time), there are little alternatives than to compute it exactly.

However, if we do have information on the ordering of the values, we can do better than that. We showed that the simple strategy 'always stay in the middle' *guarantees* a  $h(x)/d_L$ -structure approximation for  $h(x) = |\mathbf{M}| \cdot \lceil \frac{2}{c} \rceil$  in the worst case, which is the same as the expected value if we would randomly guess the values. We showed that it is NP-hard to  $h(x)/d_L$ -structure approximate MPE for  $h(x) = |\mathbf{M}| - 1$ .

The gap between these two results might leave some room for improvement. One suggestion, that we leave for future work, is to investigate whether it could help to use *monotonicity properties* in the network to get a better structure approximation.

## Acknowledgments

The author wishes to thank Iris van Rooij and Todd Wareham for helpful discussions on this topic.

## References

- [1] A. M. Abdelbar and S. M. Hedetniemi. Approximating MAPs for belief networks is NP-hard and other theorems. *Artificial Intelligence*, 102:21–38, 1998.
- [2] C. L. Baker, R. Saxe, and J. B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.
- [3] H. L. Bodlaender, F. van den Eijkhof, and L. C. van der Gaag. On the complexity of the MPA problem in probabilistic networks. In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 675–679, 2002.
- [4] N. Chater and M. Oaksford. Ten years of the rational analysis of cognition. *Trends in Cognitive Sciences*, 3:5765, 1999.

- [5] G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2):393–405, 1990.
- [6] U. Feige, M. Langberg, and K. Nissim. On the hardness of approximating NP witnesses. In K. Jansen and S. Khuller, editors, *Third International Workshop on Approximation Algorithms for Combinatorial Optimization*, volume 1913 of *LNCS*, pages 120–131, 2000.
- [7] M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., San Francisco, CA, 1979.
- [8] J. Gemela. Financial analysis using Bayesian networks. *Applied Stochastic Models in Business and Industry*, 17:57–67, 2001.
- [9] Matthew Hamilton, Moritz Müller, Iris van Rooij, and Todd Wareham. Approximating solution structure. In E. Demaine, G.Z. Gutin, D. Marx, and U. Stege, editors, *Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs*, number 07281 in *Dagstuhl Seminar Proceedings*, 2007.
- [10] F. V. Jensen and T. D Nielsen. *Bayesian Networks and Decision Graphs*. Springer Verlag, New York, NY, second edition, 2007.
- [11] D. Koller and N. Friedman. *Probabilistic Graphical Models: principles and techniques*. MIT press, Cambridge, MA, 2009.
- [12] R. Kumar and D. Sivakumar. Proofs, codes, and polynomial-time reducibilities. In *Proceedings of the Fourteenth Annual Conference on Computational Complexity*, pages 46–53. IEEE Computer Society, 1999.
- [13] J. Kwisthout. Most probable explanations in Bayesian networks: Complexity and tractability. *International Journal of Approximate Reasoning*, 52(9), 2011.
- [14] C. Lacave and F. J. Díez. A review of explanation methods for Bayesian networks. *The Knowledge Engineering Review*, 17(2):107–127, 2002.
- [15] P. J. F. Lucas, N. de Bruijn, K. Schurink, and A. Hoepelman. A probabilistic and decision-theoretic approach to the management of infectious disease at the ICU. *Artificial Intelligence in Medicine*, 3:251–279, 2000.
- [16] E. Millgram. Coherence: The price of the ticket. *Journal of Philosophy*, 97:82–93, 2000.
- [17] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, Palo Alto, CA, 1988.
- [18] S. E. Shimony. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68(2):399–410, 1994.
- [19] P. J. Sticha, D. M. Buede, and R. L. Rees. Bayesian model of the effect of personality in predicting decisionmaker behavior. In L. C. Van der Gaag and R. Almond, editors, *Proceedings of the Fourth Bayesian Modelling Applications Workshop*, 2006.
- [20] L. C. van der Gaag, S. Renooij, and P. L. Geenen. Lattices for studying monotonicity of Bayesian networks. In M. Studený and J. Vomlel, editors, *Proceedings of the Third European Workshop on Probabilistic Graphical Models*, pages 99–106, 2006.
- [21] L. C. van der Gaag, S. Renooij, C. L. M. Witteman, B. M. P. Aleman, and B. G. Taal. Probabilities for a probabilistic network: a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25:123–148, 2002.
- [22] I. van Rooij, J. Kwisthout, M. Blokpoel, J. Szymanik, T. Wareham, and I. Toni. Communicating intentions: Computationally easy or difficult? *Frontiers in Human Neuroscience*, 5(52):1–18, 2011.
- [23] I. van Rooij and T. Wareham. Intractability and approximation of optimization theories of cognition. *Journal of Mathematical Psychology*, forthcoming.

# Topical Influence on Twitter: A Feature Construction Approach

Menno Luiten

Walter A. Kusters

Frank W. Takes

*Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands*

## Abstract

In this paper we discuss the task of discovering topical influence within the online social network TWITTER. The main goal of this research is to discover who the influential users are with respect to a certain given topic. For this research we have sampled a portion of the TWITTER social graph, from which we have distilled topics and topical activity, and constructed a set of diverse features which we believe are useful in capturing the concept of topical influence. We will use several correlation and classification techniques to determine which features perform best with respect to the TWITTER network. Our findings support the claim that only looking at simple popularity features such as the number of followers is not enough to capture the concept of topical influence. It appears that more intricate features are required.

## 1 Introduction

The amount of information that is publicly available through the internet has drastically increased since the introduction of Web 2.0 [1]. Especially through online social networks [6], it has become extremely easy for users to share facts, opinions and news on any possible topic. When searching for information or news, we are confronted with a large number of information sources, from which we have to select what we believe to be correct and relevant content. Whereas before selecting sources of information was a matter of selecting certain websites, nowadays it is also a matter of selecting the correct users in a social network.

Within the online social network TWITTER [19], it is possible to follow users that are believed to produce relevant content. Such a user does not necessarily produce content which is relevant in general, but is more often only producing relevant content within a certain specific field of expertise. For example, Larry Page may be considered *influential* on the topic of internet search, but not on golf, whereas the opposite may hold for Tiger Woods. Selecting relevant users to follow on TWITTER is thus a matter of selecting users that produce relevant content on a certain *topic* (though we may ultimately be interested in multiple topics).

In this paper we will define features that can help us to determine who the influential (or authoritative) users on a certain topic are. We do this by analyzing the TWITTER social network, where we consider both the history of posted messages as well as a user's position in the social graph. Our goal is to better understand the concept of influence and to derive which characteristic features of users play a role when determining influence. In order to verify the performance of (combinations of) our features, we assume a definition of influence based on the sales funnel [3], as used by internet marketers. In this setting, a user is *influential* within the network if the links within the messages of a user are clicked on a lot by other users. As a second verification approach we consider the number of times a message has been "retweeted" by other users.

The motivation for doing this research is clear: it can help us to determine who we should definitely follow on TWITTER if we are interested in a certain topic. Also, having a list of influential users on a certain subject may be helpful to introduce new TWITTER users to build their list of people to follow based on a supplied list of interests. Additionally, it may help advertisers to select influential users who are likely able to successfully promote the advertiser's products or services. In this paper we will restrict ourselves to finding long-term authorities on a certain topic, as we will analyze multiple months of TWITTER messages.

The rest of this paper is organized as follows. First, we discuss some definitions, notations and assumptions in Section 2. After discussing related work in Section 3, we describe our sampling approach in Section 4. Next we consider a set of features for determining topical influence in Section 5, which we first filter based on effectiveness, and then apply to the TWITTER network in Section 6. Section 7 concludes.

## 2 Preliminaries

In this section we will first describe some concepts with respect to the TWITTER graph, after which we describe our main problem statement.

### 2.1 Twitter

We will be using the online social network graph  $G(V, E)$  from TWITTER as the main dataset for our research. The edges (or links)  $E$  between the users (or nodes)  $V$  within the TWITTER social graph are, contrary to many other social networks, *directed*. When a user creates a link, a task which is commonly referred to as *following*, then this user can see all messages posted by the user to whom he created a link. This construct allows us to more accurately capture the real-life concept of influence as compared to a network consisting of only undirected links where it is not clear who is interested in whom. We use  $O_x$  to denote the outlinks, i.e., the set of users followed by user  $x \in V$ , and similarly we use  $I_x$  to denote the set of users that follow user  $x$ , representing  $x$ 's inlinks.

Besides following, we will also mention several other concepts common to the TWITTER network. *Tweeting* is essentially posting a short 140-character message, referred to as a *tweet*. This message is not only visible on the profile of the originating user, but also in the *feed* of each user that follows this user. The set  $M_x$  denotes the set of messages sent by user  $x$ . A user's feed shows all messages posted by followed users. By *retweeting* we refer to a message being repeated by another user, allowing content to spread through the TWITTER network. We define the set  $R_m$  as the set of retweets of a message  $m$ . Retweeting happens for example because a user finds a message interesting and worth sharing with his followers. Referring to another user is called *mentioning*, denoted within a tweet by the symbol @, basically allowing users to direct messages to each other and have a conversation via TWITTER. In order to stress that a message is about a certain subject, so-called hashtags, denoted by the symbol #, are used. An example tweet, by user AEinstein, directed at IsaacNewton (a mention), asking about user Apple (a mention) with respect to the subject #computerscience (a hashtag), retweeted by user ScienceAcademy, is shown below:

#### AEinstein

```
@IsaacNewton what do you think of the new @Apple
product? http://bit.ly/12345 #computerscience
Retweeted by ScienceAcademy
```

### 2.2 Problem Statement

Our research focuses on the issue of determining *topical influence*. Influence, as defined by the Webster dictionary, is "the power or capacity of causing an effect in indirect or intangible ways". In our case, we will try to detect this capacity not on a global scale, but with respect to a certain topic:

**Problem Statement.** *Given a social graph of users, their connections, and posted messages, which user is most influential on a certain given topic?*

We try to answer this question by defining features which we believe describe the concept of influence. The question is then how we can measure whether or not our features are successful, which depends on our definition of when someone is influential.

Trivial ways of measuring global influence include looking at the total number of followers, or a user's position within the social graph. Furthermore, commercial websites such as Klout [13] develop metrics that have been suggested as measures of influence on TWITTER. Our definition of influence is based on the idea of the *sales funnel* [3], as used in internet marketing. This process, schematically outlined in Figure 1, traditionally describes the process of a visitor of a website from the moment he enters the website until a sale or some other action is completed. In our illustrated version of the sales funnel, social media is added prior to the visitor entering the website. The motivation for using the sales funnel is that one of the major questions in social media marketing searches for the strategy that most influences the *sales* of a company. It should however be noted that social media exposure also has strong advantages outside of the sales funnel such as brand exposure, creation of goodwill, community building and more. Our approach does not explicitly measure these benefits. We consider links in TWITTER messages as potential entry points to the sales funnel, and base our definition of influence on the number of incoming visitors in the sales funnel. We will thus consider the number of clicks on links present in TWITTER messages as a way of validating influence. As

a second validation measure of influence, we consider the number of retweets. Thus our two validation measures for determining the quality of our features, and therewith our definitions of influence, are:

**Definition 1.** *Influence within an online social network is the ability to generate clicks on posted links.*

**Definition 2.** *Influence within an online social network is the ability to generate retweets of posted messages.*

The relative value of these definitions can be inferred from their relative position in the sales funnel: the clicks of Definition 1 are closer to the end of the funnel than the retweets of Definition 2. Since we would ideally measure the effect on the end of the funnel, we value generated clicks over generated retweets.

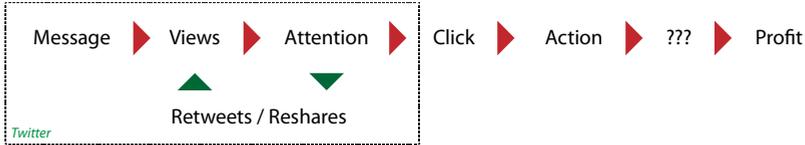


Figure 1: Schematic representation of the sales funnel.

### 3 Related Work

Trying to find the central nodes in a large network is a challenging task in the domain of data mining. Perhaps most notable is the work of Page and Brin, who introduced the well-known PageRank [14] measure for determining which nodes play an important role within the graph formed by the world wide web. Similar studies to find influential nodes have been done for social networks such as Flickr [7]. Unfortunately, the traditional PageRank-inspired measures only consider global influence, and do not take into account any topical information. Haveliwala introduced a topic-sensitive PageRank metric [10], which is applied to TWITTER through an algorithm called TwitterRank [20]. Here it is assumed that the influence of a user is a combination of the influence of his or her neighbors, and the relative amount of content of these neighbors.

With regard to influence measures, Cha et al. [5] empirically investigate the relation between common measures in influence on social media. However, when they test *topical* influence, they only take a small subset of users that have talked about all their defined topics. They find a strong correlation between topics, but this could be due to a selection bias towards generic TWITTER users, who have a tendency to talk about general topics. In other research on influence on TWITTER, specifically [2, 5, 10, 17], it is found that the traditional measures of follower counts and PageRank, while being good measures for popularity, are not as good at predicting influence when it is interpreted as the ability to engage one’s audience. Romero et al. [17] include click data in their analysis, and find a weak correlation between clicks and popularity. It is found that retweets are mostly caused by a large group of “less-connected” users, instead of particular popular TWITTER users. Wu et al. [22] have a similar conclusion and suggest that the sociological theory of two-step communication flow [11] is still valid for electronic word-of-mouth networks.

### 4 Datasets

In this section we describe our sampling method for obtaining a large TWITTER dataset for our study of topical influence. The main approach to mine the TWITTER graph is based on the Forest Fire algorithm, which was found to be a reliable method of large graph sampling by Leskovec et al. [15]. This algorithm starts by randomly selecting a user in the graph (using a numeric identifier), and retrieves all of his or her connections and profile attributes. Next, it randomly selects (“burns”)  $X$  random edges with probability 0.6, and recursively applies this step to these newly selected nodes. When the algorithm encounters an empty queue (“burns out”), it again selects a random user and repeats the process, until the required sample size is satisfied. We have used the TWITTER’s REST API for our crawling activities. We ran the crawler several times for varying amounts of time, ultimately resulting in a data sample of over 30,000 TWITTER users. Some indicative metrics on the size and shape of the data sample are shown in Table 1.

To be able to use the TWITTER graph in a topical context, we also retrieved up to 1,200 tweets for each of the users in our sample, creating the set  $M_x$  of messages for each user  $x$ . These messages were then analyzed in order to define to which topic(s) they belong, allowing us to define  $M_{tx}$ , the set of messages

Property	Value
Nodes	31, 891
Edges	584, 661
Average Degree	18.3
Modularity	0.471
Density	0.001
Clustering Coefficient	0.068
Diameter	13
Average Path Length	4.03

Table 1: TWITTER dataset characteristics.

Topic	Keywords
Politics	democratic, republican, democrats, presidential, political, election, republicans, government, federal, constitution, executive, senators, elected, congressional, representatives, politics, presidents, perry, obama, biden, gingrich, romney, santorum
Tech	web, internet, www, html, computer, data, software, online, browser, oss, opensource, programmer, programming, developer, code, coding, java, c, c#, c++, php, visual basic, python, objective-c, perl, javascript, sql, ruby, haskell, perl, actionsript

Table 2: Topics along with associated keywords.

by user  $x$  on topic  $t$ . Taking into account a cut-off value of  $|M_{t,x}|/|M_x| \geq 0.005$  to disregard users who accidentally talked about a topic, we have generated sets  $V_t \subseteq V$  of TWITTER users talking about topic  $t$ .

Our requirement for the definition of the topics was that they should be representative for a certain interest or a certain target group (e.g., politics, movies, technology, science). Using an automated topic distillation algorithm in the form of Latent Dirichlet Allocation (LDA) [4], we had no success in generating topics that complied with this requirement, as the topics more closely resembled random bags of words without a discernible theme. A similar undesired outcome was observed by [20]. Instead, we used a more simple technique of keyword matching, in which the keywords are based on the term frequency of a manually selected collection of Wikipedia articles surrounding a subject (e.g., American Politics, or Internet Technology). We empirically evaluated this distillation method to generate a more descriptive and complete set of topics compared to LDA and hashtag filtering. Throughout this paper, we use two topics in particular, namely ‘‘Politics’’ and ‘‘Tech’’, resulting in topic graphs of respectively 1, 815 and 3, 109 TWITTER users. Some keywords related to these topics are shown in Table 2.

In order to ultimately verify the influence of a user, we also gathered click data, as the number of clicks is going to serve as a measure of influence. We do this by unfolding `t.co` links that are present in TWITTER messages, and request click analytics from the ones that resolve to a `bit.ly` URL (see the example in Section 2.1). This way, we are able to retrieve a number of clicks for each link in a TWITTER message.

## 5 Features

In this section we will describe a list of features which we consider relevant with respect to topical influence, categorized based on the type of information that they use.

### 5.1 Graph-based features

Graph-based features solely consider the structure of the social graph, and are thus related only to a user  $x$ :

- The number of followers  $|I_x|$  and the number of followed people  $|O_x|$ .
- PageRank  $pr(x)$ : the most prominent measure of importance on the web [14].
- HITS authority  $a(x)$  and hub  $h(x)$  scores: an alternative measure of importance, also originally intended for the web [12].
- The 2-neighborhood  $|N_2(x)|$ : the size of the set of nodes at distance 2 from user  $x$ , extending the measure of followers by one step by counting the number of followers of followers.

### 5.2 Content-based features

Content-based features look at the message content, and are related to a user  $x$  and a topic  $t$ . We distinguish:

- The number of tweets by a user on a topic  $|M_{t,x}|$ , describing the *activity* by user  $x$  on a topic  $t$ .
- Topical ratio  $r(t, x) = |M_{t,x}|/|M_x|$ : the *relative* amount of activity of user  $x$  on a topic  $t$ , eliminating the effect of message (in)frequency.

- Term frequency-inverse document frequency  $tfidf(t, x)$ : similar to the topical ratio, but also considering the frequency of a keyword with respect to a certain topic.
- Number of mentions  $m(x) = \sum_{v \in V} |\{m \in M_v : x \text{ is mentioned in } m\}|$ . The number of times user  $x$  has been mentioned in the messages of other users can be an indication of popularity.
- Number of retweets  $rt(x) = (1/|M_x|) * \sum_{m \in M_x} |R_m|$ : this might indicate that a user or his content is popular.

### 5.3 Combined features

Considering both graph-based and content-based features, for a user  $x$  and a topic  $t$ , we can distinguish:

- Topic-sensitive PageRank  $tpr(t, x)$ : a PageRank measure that takes into account the topical ratio of the users [10].
- PageRank of a user  $x$  using only  $V_t$ , the set of users that talk about topic  $t$ , denoted  $pr(t, x)$ . This feature may indicate influence in a certain (topical) subset of users.
- Followers in the topic graph  $ti(t, x) = |I_x \cap V_t|/|I_x|$ : a high number of followers that also use the topic can indicate a topical clustering.
- Friends in the topic graph  $to(t, x) = |O_x \cap V_t|/|O_x|$ .
- Topical ratio of followers  $fr(t, x) = (1/|I_x|) * \sum_{y \in I_x} r(t, y)$ : the use of the topic by a user's followers can be indicative of a topical cluster.
- Average number of topical retweets  $rt(t, x) = (1/|M_{tx}|) * \sum_{m \in M_{tx}} |R_m|$ . This feature might not only indicate popularity on the topic, but also content value within the topical subset of users.

A more elaborate description of the features that we used can be found in [16].

### 5.4 Filtering

To determine which of the features are most relevant, we performed filtering by using Principal Component Analysis (PCA) [9] and Correlation-based Feature Selection (CfsSubsetEval) [8] from the popular data-mining software suite Weka [21]. These algorithms are designed to experiment with the feature space in order to extract the features that explain variance of the features within the dataset.

The PCA approach showed that the strongest component that was found across topic graphs consisted of *popularity features* such as HITS authority score, PageRank, the number of followers and the neighborhood size. This indicates that a large part of the variance of the features might be explained by differences in popularity. A component that was less strong, yet still significant was a component that consisted mostly of topical ratio of followers, ratio of followers in topic graph, topical retweets, topic-sensitive PageRank, etc., which we will refer to as the *topical features*. We believe this component can be interpreted to be related to the topicality of the followers of the TWITTER user.

CfsSubsetEval, contrary to PCA, recognizes a *target variable* and attempts to find a subset of features of which the composite is highly correlated with the target feature, yet uncorrelated between the selected features themselves. When targeting the number of clicks, we found that the most important features are HITS hub score, ratio of followers in topic graph, topical ratio of followers and topical retweets, as can be seen in Table 3. In this table, *merit* denotes a heuristic of the (Pearson) correlation coefficient of the subset with the target variable. This indicates a certain importance of use of topicality by both the user and the followers of the user. Interestingly, popularity measures such as followers and PageRank are only found when the number of topical retweets  $rt(t, x)$  is used as target, but not when the number of clicks  $c(t, x)$  is used. Also, during our experiments, we noticed that removing the feature of average topical retweets resulted in a significant decrease in the correlation of the subset with the target feature of average clicks.

## 6 Experiments

Using the features found as a result of the filtering process in Section 5, we have tried to find classifiers that can explain the target features using the relevant features. As the source features for the classifiers we have used the two components, popularity features and topical features, as found in the PCA step from Section 5.4. We also used the relevant features found by CfsSubsetEval, namely:

Topic	Target	Merit	Selected attributes
Politics	$c(t, x)$	0.745	$ti(t, x)$ $h(x)$ , $fr(t, x)$ , $rt(t, x)$
Politics	$rt(t, x)$	0.360	$pr(x)$ , $ti(t, x)$
Tech	$c(t, x)$	0.458	$h(x)$ , $fr(t, x)$ , $rt(t, x)$
Tech	$rt(t, x)$	0.454	$a(t, x)$ , $pr(x)$ , $ti(t, x)$ , $rt(x)$

Table 3: Results of CfsSubsetEval on topics.

- Popularity features: authority score  $a(x)$ , hub score  $h(x)$ , global PageRank  $pr(x)$ , average number of retweets  $rt(x)$  and average number of mentions  $m(x)$ .
- Topical features: ratio of topical followers  $ti(t, x)$ , follower ratio  $fr(t, x)$  and average number of topical retweets  $rt(t, x)$ .

## 6.1 Classification

Now that we have extracted the relevant features, we are ready to start our process of classifying the target attribute in a way that can explain or even predict who the influential TWITTER users are. We will do this by classification of our two target attributes  $c(t, x)$ , the number of clicks on posted links, and  $rt(t, x)$ , the number of retweets as defined in Section 2.2. Our goal is to find a classifier that is not only accurate, but also easily interpretable and understandable. As a first step we have looked at naive Bayes classifiers and C4.5 decision trees [21]. We have discretized the number of clicks into four distinct categories (class 0 through 3, from no clicks at all, to a large number of clicks) and have used Cohen’s kappa  $\kappa$  [9] as a measure of accuracy of the classifier. When the classifier finds (combinations of) features representative for certain classes of clicks, we can investigate the role of topicality of those features and interpret the classifier.

We trained classifiers on several topic graphs; the result of one topic can be seen in Table 4 ( $\kappa = 0.4465$ ) and Table 5 ( $\kappa = 0.238$ ). We noticed that only a few attributes have an increasing mean towards the higher classes of clicks, most prominently being average topical retweets, whereas most topical attributes have erratic, constant or even decreasing influence.

Because we suspected that even the filtered features were too detailed for the classification, we finally used a genetic algorithm [18] to find a combination of features that optimizes the kappa metric. While this approach may seem similar to PCA, it differs as it allows feature elimination in the classification attributes and uses the target variable for the accuracy of the model, simplifying the approach as a whole. We chose to use at least two features as a result of earlier findings from the PCA step, where we found a popularity and a topical feature set.

We again used the relevant features from Section 5 and combined them using linear weighting to generate the composite attributes, using 10-fold cross-validation to train and test the generated combinations. We experimented with adding attributes until the classifiers no longer improved their accuracy, which can be seen in Figure 2 and Figure 3. It can be observed that we only need to use a very limited number of attributes to optimize the classification of the model ( $\kappa = 0.663$ ). Interestingly, it turned out that the features that were used by the algorithm consistently were various popularity features (mentions, retweets, HITS, PageRank), but only one topical feature, namely the number of topical retweets. It turned out that excluding

Attribute	Class 0	Class 1	Class 2	Class 3
$fr(t, x)$	0.0129	0.0179	0.0201	0.0071
$a(x)$	0.0001	0.0002	0.0008	0.0019
$h(x)$	0.0001	0.0001	0.0002	0.0003
$pr(x)$	0.0001	0.0002	0.0005	0.0014
$ti(t, x)$	0.3489	0.3908	0.3829	0.2513
$rt(x)$	0.0069	0.0229	0.0831	0.2753
$m(x)$	0.0147	0.0540	0.1421	0.3464
$rt(t, x)$	0.3294	6.5952	35.649	144.43

Table 4: Mean attribute values from topic “Politics” of the naive Bayes classifier.

Component	Class 0	Class 1	Class 2	Class 3
Popularity	-0.992	0.185	2.497	6.557
Topical	0.334	0.788	0.800	-0.831

Table 5: Mean principal component values from topic “Politics” of the naive Bayes classifier.

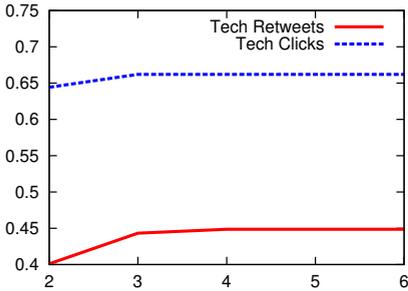


Figure 2: Kappa of best solution found (vertical axis) for increasing number of attributes (horizontal axis) on topic “Tech”.

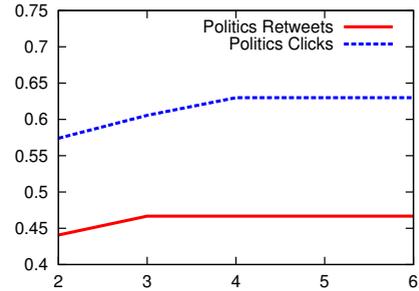


Figure 3: Kappa of best solution found (vertical axis) for increasing number of attributes (horizontal axis) on topic “Politics”.

topical retweets had a significant impact on the accuracy of the model (maximum  $\kappa$  was reduced to 0.458), demonstrating the importance of topical retweets in the classification model.

## 6.2 Discussion

We believe that our observations are in accordance with earlier work. Bakshy et al. [2] found that the number of followers does not represent influence in the spreading of messages, and that large retweet cascades are originated mostly from many “less-connected” ordinary users. Our findings show that clicks correspond to high topical retweets, supporting the statement that popularity is only a secondary feature, whereas on-topic retweets is the most dominant primary feature. Romero et al. [17] state that influence is determined by activity of followers, instead of passive attributes such as number of followers. We confirm this observation by showing that topical retweets are an activity originating from followers, and not a passive metric such as the number of (topical) followers. Cha et al. [5] also suggest that number of followers are not the most important metric of influence in both a static as well as a changing environment. Instead they propose content value as a more superior metric. We believe topical retweets are an indication of content that fits well with the user’s audience, which has been built over time, thus being a metric for both popularity, community and persistent content value.

## 7 Conclusion

Throughout this paper we have discussed various features that are useful in predicting topical influence on TWITTER. After a thorough investigation of which features contribute to predicting influence, we found two major classes of features: topical features and popularity features. Given our definition of influence based on the sales funnel, where the goal is to generate clicks on posted messages, the feature of topical retweets was found to be predominant in all classifiers. Apparently, when determining topical influence, it is most helpful to primarily investigate the interactions the user causes on his topical messages, especially regarding topical retweets. Our findings confirm earlier work which states that popularity features alone, such as the number of followers, are not sufficient to accurately capture the concept of influence.

In future work we would like to investigate if it possible to determine the extent to which a classification technique depends on the type of chosen topic. We are specifically interested in whether or not our approach works on short-term topics such as a specific soccer match or a local earthquake. Our current approach has been tested on various long-term topics and corresponding keywords, but it may very well be that when short-term topics are chosen, different classification techniques work better. We are also interested in how the influence of a user changes over time. Can we not only detect influential users, but also predict which user is going to become influential on a certain topic in the near future?

## Acknowledgments

We thank Carlos Soares and Pedro Quelhas Brito at LIAAD, University of Porto, for their suggestions regarding this project. The third author is supported by the NWO COMPASS project (grant #612.065.92).

## References

- [1] A. Ankolekar, M. Kröttsch, T. Tran, and D. Vrandečić. The two cultures: Mashing up Web 2.0 and the semantic web. In *Proceedings of the 16th International World Wide Web Conference (WWW '07)*, pages 113–114, 2007.
- [2] E. Bakshy, J.M. Hofman, W.A. Mason, and D.J. Watts. Everyone’s an influencer: Quantifying influence on Twitter. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM '11)*, pages 65–74, 2011.
- [3] T.E. Barry. The development of the hierarchy of effects: An historical perspective. *Current Issues & Research in Advertising*, 10(2):251–295, 1987.
- [4] D. M. Blei, A. Y. Ng, and M.I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [5] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in Twitter: The million follower fallacy. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM '10)*, pages 10–17, 2010.
- [6] M. Faloutsos, T. Karagiannis, and S. Moon. Online social networks. *IEEE Network*, 24(5):4–5, 2010.
- [7] A. Goyal, F. Bonchi, and L.V.S. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM '10)*, pages 241–250, 2010.
- [8] M. Hall. Correlation-based Feature Selection for Machine Learning. *PhD Thesis*, University of Waikato, 1998.
- [9] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, second edition, 2009.
- [10] T.H. Haveliwala. Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15:784–796, 2003.
- [11] E. Katz and P. Lazarsfeld. *Personal Influence: The Part Played by People in the Flow of Mass Communications*. Free Press, 1955.
- [12] J.M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
- [13] Klout, Inc. Klout. <http://www.klout.com>, accessed June 1, 2012.
- [14] A.N. Langville and C.D. Meyer. *Google’s PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006.
- [15] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pages 631–636, 2006.
- [16] M. Luiten. Topical Influence on Twitter: A Feature Construction Approach. *Master Thesis*, Leiden University, 2012.
- [17] D.M. Romero, W. Galuba, S. Asur, and B.A. Huberman. Influence and passivity in social media. In *Proceedings of the 20th International Conference Companion on World Wide Web (WWW '11)*, pages 113–114, 2011.
- [18] S.N. Sivanandam and S.N. Deepa. *Introduction to Genetic Algorithms*. Springer, 2007.
- [19] Twitter, Inc. Twitter. <http://www.twitter.com>, accessed June 1, 2012.
- [20] J. Weng, E.-P. Lim, J. Jiang, and Q. He. TwitterRank: Finding topic-sensitive influential Twitterers. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM '10)*, pages 261–270, 2010.
- [21] I.H. Witten, E. Frank, and M.A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, third edition, 2011.
- [22] S. Wu, J.M. Hofman, W.A. Mason, and D.J. Watts. Who says what to whom on Twitter. In *Proceedings of the 20th International World Wide Web Conference (WWW '11)*, pages 705–714, 2011.

# Leveraging Unscheduled Event Prediction through Mining Scheduled Event Tweets

Florian A. Kunneman

Antal van den Bosch

*Centre for Language Studies, Radboud University Nijmegen;  
P.O.Box 9103, 6500 HD Nijmegen, The Netherlands*

## Abstract

A considerable portion of social media messages is devoted to current events. Aside from references to events that recently happened, social media messages may also refer to events that have not occurred yet. Future events, such as football matches in the case study we present here, may be scheduled and known to happen; other future events, such as transfers of football players, may only be rumoured, and may in fact not happen in the end. We describe a news mining component that learns to identify tweets referring to scheduled and unscheduled future events, by being trained on messages referring to scheduled future events (as the latter are easy to harvest). Our results show that discriminating between tweets that refer to upcoming football matches and tweets that refer to past matches can be done relatively reliably with supervised machine learning methods. However, when these trained models are applied to unscheduled events, performance drops to near-baseline performance. We discuss how these results can be explained by the distinction between event type and event domain.

## 1 Introduction

Signalling the likelihood of impending events can be a valuable tool for journalists as well as for the news-reading public, who both wish to be on top of the news as it happens. The massive amount of short messages posted via the medium of `twitter.com`, so-called *tweets*, provide a potentially valuable source of information for this task, outperforming newswire articles in terms of dynamics and pluralism. A key step in the automation of this task is to be able to identify tweets posted to pass on information about, or state an opinion on, an event that has not occurred yet and that may have a high impact or news value. However, such tweets will only represent a small group within the total mass of tweets posted at a selected moment in time, making their discovery a needle-in-the-haystack problem.

One route to detect tweets referring to future events is to train a classifier on positive and negative examples of such tweets gathered from news archives with hindsight knowledge. The intuition is that tweets referring to a future event contain features distinctive from other tweets, including the closely related class of tweets referring to ongoing or past events. For example, future tense and the presence of time adverbs such as *soon* may be strong predictors for English tweets [5]. In order to create a model that captures these features and their weights, a sufficient amount of training material is needed. In this paper we set out to identify tweets referring to future scheduled and *unscheduled* events, where we collect positive cases by harvesting tweets referring to *scheduled* events we know about beforehand. Tweets of this type can often be collected with relatively little effort, as we will demonstrate for the case study domain of football<sup>1</sup>. Scheduled events are often marked by a predictable hashtag, the common way to mark an explicit keyword in a tweet by adding a '#' before a word, that is either recommended in a top-down fashion or has become conventionalized over time. In contrast, hashtags referring to unscheduled events tend to emerge during the process and can have various unpredictable forms.

Although all processed tweets will be embedded in the domain of football, it is not certain whether training on the temporal nature of tweets referring to specific football matches will be effective for the classification of another event type within the same domain such as football transfers, let alone events in

---

<sup>1</sup>We use the term 'football' as the historically accurate name for the sport that is sometimes referred to as 'soccer'.

other domains. This paper describes a case study aimed to test to what extent the similarity between tweets referring to future events can be leveraged across tweet types in the same domain. Classifiers are trained on tweets referring to football matches in the Dutch league and tested on tweets referring to other sorts of matches and unscheduled transfers of football players from one team to another that may or may not materialize. With this research we aim to find out whether the almost effortless collection of training material based on forward knowledge of scheduled events is beneficial for the detection of anticipating tweets in other domains, and ultimately the set of all tweets posted.

This paper is structured as follows. In Section 2 we provide an overview of the relatively large body of recent work on event detection in social media; we review the common trends in this field and zoom in on related work aimed at detecting future events. Section 3 introduces the domain of the case study: scheduled and unscheduled football events. In Section 4 we describe our series of experiments and their results on classifying tweets on football matches into tweets referring to future events versus present or past matches, and on classifying tweets on football transfers. We summarize, state our conclusions, and formulate points for further research in Section 5.

## 2 Related Work

The idea that messages in social media can be used as a source for the prediction of a future event or outcome has been explored in a number of studies. [2] aim to predict the commercial success of specific movies based on the number of tweets that refer to the movie from a week before the premiere. Furthermore, they perform automatic sentiment analysis on tweets posted in the first week after release. [10] perform trending news detection to improve the prediction of stock market changes. [9] aim to predict whether future events mentioned in tweets will actually occur by discovering events that have a causal relationship. Such event pairs were mined from news archives by searching for certain lexical causality connectors in titles, and normalized by extracting verbs and nouns and connecting them to an ontology. Although these studies consider tweets that refer to future events, the automatic detection of tweets expressing the anticipation of future events has not been investigated to the best of our knowledge.

In order to collect tweets regarding events from the total stream of available tweets, irrelevant messages such as conversational tweets and tweets aimed to share personal experience should be filtered out first. [12] tackle this problem by classifying tweets as either *junk* or *news* based on training on a handlabeled set of tweets, and thereby collect suitable data for a news processing system. Instead of filtering, one can also focus on the distribution of topics discussed on twitter in time, and thereby dispose of tweets not referring to news events if they can be identified as a topic. [8] apply first story detection (the emergence of a news event from a first mention onwards) in tweets, where major events are detected as chains of tweets linked by a similarity score. This way new topics that have a certain significance are detected online. Rather than filtering news tweets from spam or paying attention to topics, tweets linked to events could also be detected by looking at their linguistic structure. [5] try to extract future events referred to in tweets by searching for specific patterns such as phrases consisting of a verb in the future tense combined with the mention of a time expression.

The detection of tweets referring to events tends to become simpler when the domain of the events searched for is restricted. [11] are interested in tweets mentioning an earthquake in order to warn endangered residents in an early stage. The target tweets are detected by simply searching for tweets with the word 'earthquake'. [6] describe a service to monitor specific events, where the domain is based on user input. The input is enriched by semantic ontologies, thereby filtering the interesting tweets and creating a network around the event. [1] have created *twitcident*, a service to follow current emergencies. The tweets collected before additional filtering are retrieved by keyword search based on input from a police communication network on which emergency services immediately broadcast incidents.

The collection and filtering of tweets referring to *planned* or *scheduled* events is a goal in several studies. [3] aim to provide users with a service to seek information about different stages of the scheduled event (before, during and after the event). They base the keywords connected to events on information from sites such as *upcoming.com*. In order to collect the right tweets, keywords are restricted to a location and specific words describing the event. Additionally, the results from over 50 event queries were labeled by hand, and high precision tweets were used to define new queries and retrieve additional event messages. [4] retrieve tweets referring to matches in the cricket world cup during time of play, and try to extract descriptions of specific micro-events (such as a player scoring a wicket). The tweets are collected during match time using keywords based on general references to the world cup and on common terminology in

the domain of cricket. [7] are also interested in events during matches, focusing on football and rugby, and want to automatically provide the end user with highlights of a match in the form of short segments from the live coverage. Tweets referring to events in matches are collected by queries composed of keywords consisting of the first three letters of the competing teams (not concatenated) and a keyword with reference to the league or cup in which the match is played.

### 3 Case study: Football Events

The case study described in this paper concerns the classification of tweets referring to football matches as scheduled events and football transfers as unscheduled events. The goal is to test if there is an overall pattern in anticipating tweets, i.e. tweets that refer to future football events. A practical reason why Dutch tweets in the domain of football are collected as target material is that football is the number one sport in the Netherlands, and accordingly a sufficiently large number of people tweet about football matches and transfers. Furthermore, there is a multitude of events in the form of matches each round of the league, enabling a lot of keyword-based searching. A key advantage of tweets referring to scheduled events is that it can be established exactly, by their time stamp and the known timing of the scheduled event they refer to, whether they are posted before, during or after a match.

## 4 Experiments

Our case study consists of two experiments. In the first experiment, described in Section 4.1, we train supervised machine-learning classifiers to distinguish before-match tweets from tweets generated during or after a match. In the second experiment, described in Section 4.3, a classifier trained on the former type of tweets is applied to tweets referring to transfers, testing if tweets referring to scheduled events can be useful training material for determining whether a tweet is mentioning a future unscheduled event.

### 4.1 Football Matches: Experimental Setup

#### 4.1.1 Corpus

The corpus used in this study consists of tweets referring to football matches. The tweets are collected in an online fashion by means of selected search terms. The convention to refer to a football match by concatenating the first three characters of the home and away team respectively to serve as hashtag (for example, '#ajafey' for home team Ajax playing against Feyenoord) was used for high-precision retrieval of match tweets. All matches in the Dutch premier league, the *Eredivisie*, were harvested through these conventional match hashtags, and collected in the period from April 3, 2012, until May 23, 2012, the final weeks of the 2011–2012 season, including play-offs (small tournaments to settle promotion / relegation or tickets to European football). In addition we collected tweets referring to the UEFA Champions League final between Bayern Munich and Chelsea FC of May 19, 2012. The retrieved tweets were restricted to Dutch user accounts in order to maintain a single language throughout the tweets as much as possible. The tweets were collected by sending queries to the Twitter API every two minutes. This short interval was applied in order to catch all the tweets posted during matches, as there is a high density of football-related tweets during game time. The keywords in the form of match hashtags were all linked to a specific timeslot in which the match was played, in order to directly label each incoming tweet based on the time at which it was posted ('before', 'during' or 'after' a match).

The resulting set of tweets was filtered by removing duplicate tweets, retweets and tweets that only consisted of a url or hashtag. This resulted in a final set of about 70 thousand tweets. These tweets were tokenized by *ucto*, a rule-based tokenizer for Dutch<sup>2</sup>. The tokenized tweets were additionally cleared of punctuation, URLs and hashtags.

#### 4.1.2 Classification

In order to obtain a broad sense of machine-learning classifier performance on the task, five typical supervised classification algorithms were applied:  $k$  nearest-neighbor (Knn) classification, Winnow, SVM,

<sup>2</sup><http://ilk.uvt.nl/ucto>

Single words ( <i>English translation</i> )	Expressions
morgen ( <i>tomorrow</i> )	volgende? (dag week maand weekend)
overmorgen ( <i>the day after tomorrow</i> )	(aan)?komende? (dag week maand weekend)
straks ( <i>soon</i> )	(kaartje ticket)s?
binnenkort ( <i>soon</i> )	
zo direct ( <i>soon</i> )	
zometeen ( <i>soon</i> )	
zin in ( <i>look forward to</i> )	
maandag-zondag ( <i>Monday-Sunday</i> )	

Table 1: Marker words and expressions on which the vocabulary baseline is based.

MaxEnt, and Naive Bayes. For Knn and SVM, the *PyML*<sup>3</sup> implementation was used, while the MACHine Learning for Language Toolkit<sup>4</sup> was used for Winnow, Naive Bayes and MaxEnt classifiers. SVM was applied with a second-order kernel and the  $k$  hyperparameter of Knn was set to 5.

The different event subdomains distinguished in the retrieved football tweets were league matches, play-off matches, and the 2012 Champions League final. Distinguishing features of matches in the play-offs are the definite character (there is more at stake than in the case of most league matches) and the fact that a smaller pool of clubs is involved. This might result in more emotional tweets and tweets from a more specific group of people supporting the clubs. The Champions League final shares the definite character with a high chance of emotionally loaded tweets. On the other hand, a broader public in comparison to league and play-off matches is compelled to tweet about the final. The total set of retrieved tweets contains 57,109 tweets referring to one of 86 league matches, 7,382 tweets referring to one of 20 play-off matches, and 3,404 tweets referring to the Champions League final. 10-fold cross-validation was performed on the league tweets in a first series of experiments. Then, all league tweets were used as a single training set for the classification of the tweets referring to the play-offs and the final.

The selection of features was kept to words only: from the word sequences in the tweets we derived unigram, bigram, and trigram features. By combining the three sorts of  $n$ -grams, bonuses are awarded to matching on longer  $n$ -grams, on top of the weights that their underlying unigrams already represent. For example, the most frequent trigram is ‘voor de wedstrijd’ (*before the game*). While ‘voor’ (*before*) and ‘de wedstrijd’ (*the game*) are informative features in their own right, the combination adds its own discriminative power. Dimensionality is restricted by pruning all features occurring less than ten times. In a preliminary classification run the removal of stopwords, stemming, lemmatization and the addition of part-of-speech tags did not lead to an improvement of the results. A possible explanation for this is that tweets contain rather non-standard language and tokens, and linguistic preprocessing is therefore unreliable. For this reason no additional preprocessing was used for the experiments leading to the following results.

## 4.2 Football Matches: Results

The results of classification on tweets referring to football matches are listed in Table 1. Results are given in terms of precision, recall, and F1-scores of the identification of ‘before’ tweets. The ‘before’ baseline refers to the baseline strategy of labeling all tweets as ‘before’. The ‘marker words’ baseline consists of classifying all tweets as ‘before’ that contain one of a manually created set of words or expressions that mostly refer to future time. The set of marker words and expressions is displayed in Table 2.

The table shows that all five classifiers obtain a reasonable precision and recall for league match classification, scoring between 0.1 and 0.15 above the ‘before’ baseline F1 result. When classifying play-off tweets based on league training data, the improvement over the ‘before’ baseline F1 is smaller, both due to a lower precision and recall. The performance on tweets referring to the Champions League final is worse, but still better in comparison to the baseline. The ‘before’ baseline for this subset is quite low as the percentage of tweets posted before the final is a lot smaller than in the case of the other subsets. The ‘marker words’ baseline consistently leads to the best precision, but performs bad in terms of recall. This shows that the set of ‘before’ tweets in each domain has quite some diversity, and literal future references are but one type of indicator. The precision scores of the classifiers indicate that the league tweets as training data do help in

<sup>3</sup><http://pyml.sourceforge.net/>

<sup>4</sup><http://mallet.cs.umass.edu/>

	league matches (10-fold)			play-off matches			CL final		
	precision	recall	F1	precision	recall	F1	precision	recall	F1
‘before’ baseline	0.57	1	0.73	0.56	1	0.72	0.38	1	0.55
‘marker words’ baseline	0.92	0.22	0.36	0.90	0.22	0.36	0.81	0.15	0.26
Naive Bayes	0.86	0.88	0.87	0.76	0.83	0.79	0.57	0.86	0.69
MaxEnt	0.88	0.88	0.88	0.83	0.75	0.79	0.69	0.8	0.74
Winnov	0.77	0.88	0.82	0.78	0.67	0.72	0.58	0.68	0.62
SVM	0.88	0.88	0.88	0.82	0.76	0.79	0.6	0.77	0.68
Knn	0.76	0.89	0.82	0.67	0.79	0.73	0.46	0.84	0.60

Table 2: Precision, recall and F1-scores on labeling tweets as ‘before’ in three experiments: 10-fold cross-validation on league matches (left), on the post-season playoff matches (middle), and on the 2012 Champions League final (right).

	league matches (10-fold)			play-off matches			final		
	precision	recall	F1	precision	recall	F1	precision	recall	F1
‘before’ baseline	0.75	1	0.86	0.75	1	0.86	0.45	1	0.62
‘marker words’ baseline	0.94	0.22	0.36	0.92	0.22	0.36	0.84	0.15	0.26
Naive Bayes	0.88	0.89	0.89	0.82	0.95	0.88	0.57	0.96	0.71
MaxEnt	0.88	0.94	0.91	0.87	0.89	0.88	0.71	0.88	0.79
Winnov	0.84	0.90	0.87	0.84	0.85	0.85	0.66	0.75	0.7
SVM	0.91	0.94	0.92	0.86	0.86	0.86	0.58	0.85	0.69
Knn	0.84	0.96	0.9	0.79	0.91	0.85	0.48	0.94	0.64

Table 3: Precision, recall and F1-scores on labeling tweets as ‘before’ in league matches (left), playoff matches (middle), and the CL final (right), with game-time (‘during’) tweets removed.

distinguishing tweets anticipating the final, with a markedly higher precision for both SVM and MaxEnt in comparison to the other classifiers.

Because sports matches themselves are a special kind of timed event with many particular micro-events that may be the subject of messages, the set of tweets posted during a match could hamper the task of classifying ‘before’ tweets, as these tweets do not as much refer to the event at large as ‘before’ or ‘after’ tweets do. Furthermore, tweets referring to unscheduled football events will not contain these specific game-time tweets. To measure the effect of this particular class which was included in the first experiment, we performed a second experiment on an alternate version of the league, off-season and final game tweets: without the tweets posted during matches. The results of this experiment are displayed in Table 3.

The results in Table 3 indicate that removing game-time (‘during’) tweets leads to an overall improvement in the performance of the classifiers on the ‘before’ class, while on the other hand the difference with the ‘before’ baseline score has decreased. This can be explained by the fact that the relatively higher percentage of tweets with the label ‘before’ leads to a considerable improvement of baseline precision and F1. This somewhat trivial result is furthermore coloured by the fact that the removal of ‘during’ tweets can only be done in situations in which the exact game time is known, which in our training data is the case, but which may very well be unknown in another automatic news mining scenario.

When comparing the performance of the different classifiers on this dataset (both with and without tweets during matches), a number of observations can be made. In terms of F1 performance, the MaxEnt classifier has the best performance on the playoff and final subsets, suggesting that it learns the best generalizing feature weights from the league data during training. This contrasts with the SVM performance, which is strong in the 10-fold cross-validation experiments on the league data, but falls below the performance of MaxEnt on the tweets referring to the final. Knn and Naive Bayes both attain relatively high recall rates, at the cost of a lower precision. With a majority of tweets in the training data labeled ‘before’, the high value of  $k = 5$  in the Knn classifier and the high prior probability for the class lead to a high recall and low precision on the class with both algorithms.

In sum, this first experiment showed that tweets before football matches could quite accurately be distinguished from tweets after matches based on their content, and that a reasonable performance is maintained when applying the classifiers on matches of somewhat different types, without additional training. The goal of the second experiment described in the following section is to test whether training on league matches

is still valuable when applying the classifiers on the more distant event type of unscheduled transfers of football players.

### 4.3 Football Transfers: Experimental Setup

#### 4.3.1 Corpus

As the first step in collecting transfer tweets a number of rumoured transfers in Dutch professional football from the summer of 2011 until the end of the 2011–2012 season were collected from the Dutch website [www.transferboulevard.nl](http://www.transferboulevard.nl). On this site visitors can post a transfer rumour, as well as assess already posted rumours on their credibility. Every transfer rumour collected from this site contains a headline, a text, its author and assessment scores. In order to formulate a query for the collection of tweets, named entities were extracted from the headlines by means of Named Entity Recognition performed by Frog, a freely available morpho-syntactic text analyzer for Dutch<sup>5</sup>. When at least a person and an organization were identified in a headline, all named entities collectively formed a query for tweets. The idea is that the combination of a player, a new club and a time frame around the moment when the transfer either happens or fails forms an accurate set of keywords via which a collection of the tweets referring to a transfer can be harvested.

Before collecting tweets based on the formulated queries, the transfer events on which the queries were based were manually labeled as leading either to the eventual occurrence of the transfer or to the transfer not taking place, based on fact checking in reliable news sources. Rumours of transfers that still ‘slumbered’ (i.e. were not resolved at the time of writing) were removed from the set. This resulted in 90 transfer events with the label ‘occurred’ and 192 transfer events with the label ‘not occurred’.

The transfer events from which queries were formulated dated back to July 2011. The API offered by [twitter.com](http://twitter.com) does not go back this far. In order to collect all tweets in time referring to a transfer, Topsy search<sup>6</sup> with a searchable collection of past tweets from May 2008 onwards was queried using the Otter API<sup>7</sup>. This resulted in 3,852 tweets in the category ‘occurred’ and 3,731 tweets in the category ‘not occurred’, resulting in a set of 7,583 tweets in total.

#### 4.3.2 Classification

In order to evaluate the automatic classification of tweets in the collected set by the classifiers described in the previous section, all transfer tweets are labeled ‘before’ or ‘after’ by their known date of the actual occurrence or failure of the transfer. Tweets posted on the same date as a transfer outcome are given the label ‘after’, because they mostly are a reaction to the outcome of the transfer it refers to. As the outcome of a transfer might influence the tweets, this categorization is maintained for classification. Tweets referring to rumoured transfers that neither have a positive or negative outcome (slumbering rumours) are withheld from the corpus. This results in three sets of tweets on which classification is performed: tweets referring to transfers that occur, transfers that fail, and the former sets combined.

The tweets are preprocessed in the same way as the match tweets, and again the unigrams, bigrams, and trigrams from each tweet are retrieved as features. The five classifiers applied in the former experiment are trained on the league training data without game-time tweets.

### 4.4 Football transfer Results

The results of the classification are given in Table 4. The baseline score is computed on grounds of classification of all tweets as ‘before’. For the ‘marker words’ baseline, the same list presented in tabel 1 is used.

The results show a marked decline in comparison to the classification of tweets referring to matches reported in the previous section. The classifiers do not outperform the ‘before’ baseline in terms of the F1 score. The generalization performance of the MaxEnt classifier is now the lowest, while it was the best generalizing classifier of tweets regarding the final or play-off matches. In terms of recall, the Knn and Naive Bayes classifiers still retain a good performance, at the cost of a near-baseline precision. The ‘marker words’ scores show an even more marked decline, while still retaining the best precision scores. The very

<sup>5</sup><http://ilk.uvt.nl/frog>

<sup>6</sup><http://topsy.com/>

<sup>7</sup>[otter.topsy.com/](http://otter.topsy.com/)

	<b>all transfer tweets</b>			
	precision	recall	F1	accuracy
'before' baseline	0.5	1	0.67	0.5
'marker words' baseline	0.63	0.02	0.04	0.5
Naive Bayes	0.42	0.48	0.45	0.41
MaxEnt	0.39	0.34	0.4	0.39
Winnow	0.46	0.58	0.51	0.45
SVM	0.54	0.56	0.55	0.53
Knn	0.51	0.92	0.66	0.51

Table 4: Performance scores, including accuracy, on labeling transfer tweets as 'before', by the five machine-learning algorithms.

	<b>successful transfers</b>			<b>failed transfers</b>		
	precision	recall	F1	precision	recall	F1
'before' baseline	0.39	1	0.56	0.62	1	0.77
'marker words' baseline	0.57	0.04	0.07	0.78	0.01	0.03
Naive Bayes	0.56	0.76	0.65	0.41	0.7	0.52
MaxEnt	0.56	0.61	0.58	0.4	0.57	0.47
Winnow	0.53	0.68	0.6	0.38	0.63	0.48
SVM	0.41	0.51	0.46	0.62	0.6	0.62
Knn	0.39	0.93	0.55	0.62	0.92	0.74

Table 5: Performance score on labeling transfer tweets as 'before', split on successful transfers and failed transfers.

low recall indicates that the manually selected forward-referring time expressions used in this baseline are not used a lot in tweets referring to football transfers. Presumably, these tweets are characterized more by generic comments, opinions, and rumour statements than by specific pointers to the moment of a transfer.

As an extra analysis with hindsight knowledge, the classification on the different outcomes of a transfer (success or failure) are displayed in Table 5. A main difference between these two outcomes is the percentage of tweets before the conclusion of a transfer: 39% for transfers that did materialize, versus 62% for transfers that did not. Apparently, successful transfers evoke more reactions afterwards than transfers that are cancelled. When looking more closely at classifier performance, there is somewhat of a split between Naive Bayes, MaxEnt and Winnow on the one hand, performing reasonably well on occurred transfer tweets, and SVM and Knn on the other hand doing well on failed transfer tweets. Of course, the outcome of a transfer is not known in advance, so it is hard to make any conclusions based on this difference.

On the whole, the generalization performances of the classifiers applied to transfer tweets are quite low in terms of precision, recall and accuracy, underlining the difficulty of the task to classify the state of a tweet linked to an event type different from the training data, even though they are all football events.

## 5 Discussion

The case study presented in this paper shows that the period in which a tweet is posted related to an event, when discretized into 'before' and 'not before', can be classified reasonably accurately on the basis of training data with the same event type, regardless of slight event type variations, as we showed with the league matches, the play-off matches, and the Champions League final, which were all classified well when trained just on league matches. However, classifying tweets on another event type (transfers of football players) based on the same training data leads to a poor performance. Thus, the presumed similarity between anticipating tweets regardless of the event type is not apparent.

As the identification of anticipating tweets in general is an interesting task for news mining systems, more research could be undertaken starting from our current experimental setup. Instead of a case study in one domain, a more general approach may be followed in which the overall anticipating pattern is sought by collecting and performing training on tweets from many domains and event types mixed together. Al-

ternatively, more generic classifiers could be trained by explicitly filtering away event-specific and domain-specific features such as named entities and other content words, while selecting or placing more weight on tense markers and time expressions.

Another research question to be pursued in further research would be what the most discriminative factors are that characterize tweets referring to scheduled events versus those referring to unscheduled events. That it is unknown whether the latter event will happen or not is likely to add a speculative aspect to the tweets anticipating such events. The difficulty remains, however, that it takes considerably more effort to accurately harvest and label tweets anticipating unscheduled events than the virtually effortless harvesting and labeling carried out in our study.

## References

- [1] F. Abel, C. Hauff, G. Houben, K. Tao, and R. Stronkman. Semantics + Filtering + Search = Twitcident, Exploring Information in Social Web Streams. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media, HT 2012*, 2012.
- [2] S. Asur and B. A. Huberman. Predicting the Future with Social Media. 2010.
- [3] H. Becker, F. Chen, D. Iter, M. Naaman, and L. Gravano. Automatic Identification and Presentation of Twitter Content for Planned Events. 2011.
- [4] S. Choudhury and John G. Breslin. Extracting Semantic Entities and Events from Sports Tweets. May 2011.
- [5] Alan Jackoway, Hanan Samet, and Jagan Sankaranarayanan. Identification of live news events using Twitter. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks, LBSN '11*, pages 25–32, New York, NY, USA, 2011. ACM.
- [6] P. Kapanipathi, C. Thomas, Pablo N. Mendes, and A. Sheth. Continuous Semantics: Dynamically Following Events. In *Proceedings of Manufacturing & Service Operations Management (MSOM), 2011*, 2011.
- [7] J. Lanagan and A. F. Smeaton. Using Twitter to Detect and Tag Important Events in Live Sports. In *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.
- [8] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to Twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10*, pages 181–189, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [9] K. Radinsky, S. Davidovich, and S. Markovitch. Learning causality for news event prediction. In *Proceedings of the 21st international conference on World Wide Web*, 2012.
- [10] J. Ritterman, M. Osborne, and E. Klein. Using Prediction Markets and Twitter to Predict a Swine Flu Pandemic. In *1st International Workshop on Mining Social Media*, 2009.
- [11] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web, WWW '10*, pages 851–860, New York, NY, USA, 2010. ACM.
- [12] Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. TwitterStand: news in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '09*, pages 42–51, New York, NY, USA, 2009. ACM.

# Authorship Disambiguation and Alias Resolution in Email Data

Freek Maes

Johannes C. Scholtes

*Department of Knowledge Engineering  
Maastricht University, P.O. Box 616, 6200 MD Maastricht*

## Abstract

Given a data set of email messages we are interested in how to resolve aliases and disambiguate authors even if their names are misspelled, if they use completely different email addresses or if they deliberately use aliases. This is done by using a combination of string similarity metrics and techniques from authorship attribution and link analysis. These techniques are combined by using a voting algorithm that is based on a Support Vector Machine. The approach is tested on a cleaned subset of the ENRON email data set. The results show that a combination of Jaro-Winkler email address similarity, Support Vector Machine on writing style attributes and Jaccard similarity of the link network outperforms the use of each of these techniques separately.

## 1 Introduction

In this paper a description will be given of a new approach to the problem of disambiguating authorship and resolving aliases in email data. The techniques that are commonly used for authorship detection in literary texts cannot readily be applied to email data for a number of reasons. (1) The number of potential authors in an email data set can be very large, whereas traditional authorship attribution problems only deal with small author sets (2) email data is often sparse and can be very noisy because of the presence of forwards/replies, duplicates and system messages. (3) the written text contained in an email message can be very short, making it hard to distill style markers from it and (4) it is not known whether a particular person in the data set uses any aliases at all, so the candidate set is an open set.

A number of approaches exist to determining authorship of email data:

- Using *string similarity metrics*, such as Jaro-Winkler [11], on the email addresses it is possible to quickly generate a list of potential aliases of an author. These string metrics are able to capture superficial aliases that results from the use of different email domains/protocols (e.g. home or work email) and spelling errors. However, these metrics often give false positive aliases, such as "John Barker" and "John Baker" which might actually be two different persons. Moreover, they fail to find the more sophisticated aliases where the email addresses do not look alike, such as "Bin Laden" and "The Prince".
- *Authorship attribution* techniques can be used to find the author of a given email solely by looking at the writing style that a particular author employs. By training a binary classifier on a combination of lexical, syntactic, content-specific and/or semantic features derived from training messages, it is possible to determine the author of a new anonymous message. Multiple classifiers can then be combined using a one-versus-all approach such that a multi-class problem can also be solved.
- The information that is captured in the *link network* of the author can also be utilized. For example: if two authors share a great number of direct contacts the likelihood that they might be the same person increases. Similarly, information from more distantly shared contacts can be used to provide additional information about the similarity between two persons.

Features	Description
<b>Lexical</b>	
1	Total number of characters (C)
2	Total number of alphabetic characters / C
3	Total number of upper-case characters / C
4	Total number of digit characters / C
5	Total number of white-space characters / C
6	Total number of tab spaces / C
7-32	Frequency of letters A-Z
33-53	Frequency of special characters ~@#%&*-_+=<[ ] { } \
54	Total number of words (M)
55	Total number of short words / M less than four characters
56	Total number of characters in words / C
57	Average word length
58	Average sentence length (in characters)
59	Average sentence length (in words)
60	Total different words / M
61	Hapax legomena Frequency of once-occurring words
62	Hapax dislegomena Frequency of twice-occurring words
63-82	Word length frequency distribution / M
83-333	TF*IDF of 250 most frequent 3-grams
<b>Syntactic</b>	
334-341	Frequency of punctuation , . ? ! ; ' ”
342-491	Frequency of function words
<b>Structural</b>	
492	Total number of sentences

Table 1: Feature set that has been used in the authorship SVM

Since the three approaches mentioned above use information from different domains, the hypothesis of this research is that combining them will yield better results than each technique on its own. In order to combine them, a separate binary classifier on the results of the three methods has been trained that can distinguish between good and bad combinations of results.

## 2 Approach

*Authorship based on email address similarity:* Christen [3] found that when dealing with surnames the Jaro similarity metric performed best out of 27 techniques. Cohen and Fienberg [5] evaluated different string metrics on different data sets and found that the Monge-Elkan distance performed best. However, they conclude that the Jaro-Winkler metric performed almost as well as the Monge-Elkan distance, but is an order of magnitude faster. Therefore, in the experiments to follow, a Jaro-Winkler similarity has been calculated between each author-candidate pair based on their email addresses. The Jaro distance calculates the similarity between two strings based on the number of matching characters, and the number of transpositions needed to transform one string into the other. The Winkler-enhancement increases the Jaro-score when the two strings share a common prefix. This approach will hereafter be referred to as "Jaro-Winkler" or "JW".

*Authorship based on content:* For every email message a combination of lexical, syntactic and structural features has been extracted. Examples of features that have been used are word and character frequencies, frequency of punctuation, vocabulary richness measures, 3-grams, sentence length and frequency of function words. The complete list of features, partially adapted from [12] and extended with a number of additional feature to create a larger overall feature variance, can be found in table 1. For each author a Support Vector Machine (SVM) has been created using the author's email as positive, and a random selection of other emails as negative training examples. Both classes have been balanced in the number of training instances. This approach will be referred to as "authorship SVM".

*Authorship based on link-analysis:* Two link analysis methods have been employed in order to detect

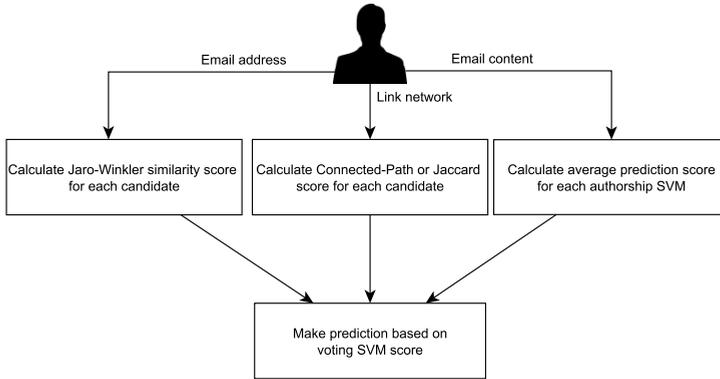


Figure 1: The structure of the framework.

aliases in the link network. The first one is the well-known Jaccard similarity [8], which will be referred to as "Jaccard". Let  $v, w$  be two authors in the data set and  $N(v), N(w)$  the direct neighbors of  $v$  and  $w$  respectively. The similarity between  $v$  and  $w$  is calculated as follows:

$$Jaccard(v, w) = \frac{|N(v) \cap N(w)|}{|N(v) \cup N(w)|}$$

The second link analysis method is a more sophisticated method referred to as "Connected Path" or "CP". Connected Path [1] has been shown to outperform a range of well-known algorithms and metrics that can be used for alias detection in link networks, such as Jaccard similarity, Connected Triples, Pagerank and PageSim. The Connected Path-algorithm values shorter paths between authors higher than longer ones. Moreover, the more connections an author has to other authors, the lower each connection is valued. By aggregating in a smart way over all possible paths between two authors the algorithm derives a similarity metric that indicates how similar the two authors' link networks are.

*Combining the results using an SVM voting algorithm:* The results of these different techniques were then used to train a separate SVM. This SVM will be referred to as the "Voting SVM". Since the SVM performs feature ranking internally, it automatically assigns weights to different combinations of results and can distinguish between successful and unsuccessful combinations of results. If the results of one technique are ambiguous, another technique can possibly aid in making the classification decision. The general structure of the framework is summarized in figure 1. Two combinations of techniques have been tested, namely Jaro-Winkler, Connected Path similarity and authorship SVM ("JW-SVM-CP"), and Jaro-Winkler, Jaccard similarity and authorship SVM ("JW-Jaccard-SVM"). In order to avoid over-fitting, the voting SVM is trained on instances that did not occur in the test set.

*The ENRON Data set:* The new approach has been tested on the ENRON-data set that was made available by the Federal Energy Regulatory Commission during its investigation into fraudulent activities at ENRON [7]. A well-known version of the data set, containing roughly 500,000 email messages from 151 Enron employees, was first made available by William Cohen [4]. Later, Shetty & Adibi [10] applied preprocessing such as removing empty messages and duplicates to the data set. The Shetty & Abidi-version of the data set has been used in this research. Many records in this data set consisted of system messages, emails with little or no original text (e.g. forwards or empty messages) and duplicates. These messages have been removed in order to reduce noise. Messages where the number of words (after removing forwarded information) was smaller than or equal to 10 were also removed, since they contained too little useful information.

According to Burrows [2] 10,000 words per author is a reliable minimum for authorship attribution, whereas Sanderson and Guenter [9] mention a minimum of 5,000 words per author. Since Hirst and Feiguina

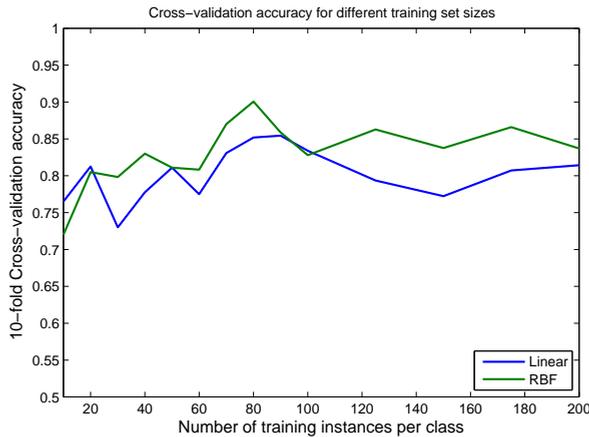


Figure 2: Averages of 10 times 10-fold cross-validation using different training set sizes and kernels for the authorship SVM.

[6] conclude that using multiple short texts for authorship attribution overcomes the problem of not having sufficiently long training texts available, there is no need to concatenate the e-mails from a single author into one long e-mail. Empirical findings on the ENRON data set, displayed in figure 2, show that using 80 training instances per class in combination with a Radial Basis Function-kernel (RBF) achieves the highest accuracy. Therefore, it was decided that a RBF-kernel should be used and that authors with a total number of emails less than 80 should be discarded. Additionally, in order to preserve balance in the number of training instances per author, authors that had sent more than 600 messages were also removed from the data set. In the final data set the average number of words per email equals 209, and with at least 80 emails per author it is ensured that each author has a reliable number of words to train on. After preprocessing the data set consisted of 44,912 emails by 246 different authors.

*Training and Evaluation:* Since there was no data to verify whether the ENRON-data set actually contained any real aliases, authors that had a total of more than 200 messages were split up into aliases of 100-200 messages each. For each author with more than 200 messages there were two possibilities:

- The author is split up into 1 or more artificial aliases yielding high Jaro-Winkler similarity. These are easy-to-recognize aliases, for example: *john.doe@enron.com* is split up into the aliases *john.doe@enron.comA* and *john.doe@enron.comB*.
- The author is split up into 1 or more artificial aliases yielding low Jaro-Winkler similarity. These are hard-to-recognize aliases, for example: *jane.doe@enron.com* is split up into the aliases *bin\_laden* and *abu\_abdallah*.

In total, 41 authors were split up into aliases with high Jaro-Winkler similarity, and 12 authors were split up into aliases with low Jaro-Winkler similarity. Emails from and to the original authors were randomly assigned to one of the author's artificial aliases, and separate authorship SVM's were trained for each alias. Splitting up authors may result in aliases with the same e-mail signatures (name, position, telephone number, etc.). However, there are not many emails in the dataset that contain such an extensive signature. In addition, the content-based approach is not affected by this since it does not take into account the most frequent n-grams per author, but the most frequently occurring n-grams in the complete data set. In order to evaluate the results of the different techniques two different test sets have been created, both of which can be seen in table 2. The first test set, called the *mixed* test set, has a fairly equal division of alias types. The second test set, called the *hard* test set, is substantially more difficult since the majority of the aliases are not easy to recognize by their email addresses. The authors in each test set were chosen at random from their respective alias categories.

Test set:	Mixed	Hard
High Jaro-Winkler	6	2
Low Jaro-Winkler	8	16
No alias	6	2

Table 2: Distribution of alias-types for two different test sets.

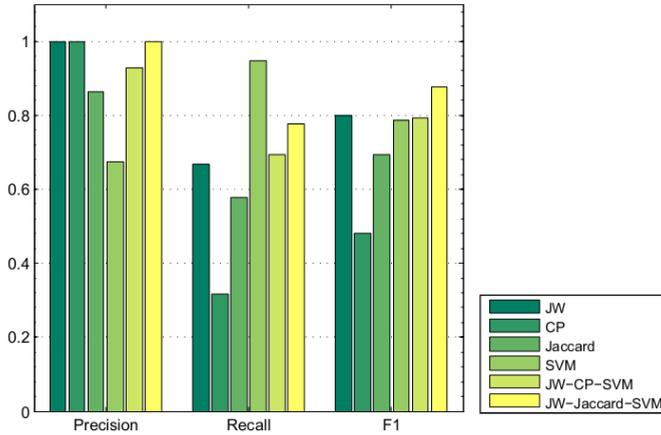


Figure 3: Precision, Recall and F1-scores for different techniques, evaluated on the mixed test set.

### 3 Results

Figure 3 gives an overview of the precision, recall and F1-values that correspond to the best F1-score for each technique on the mixed test set. Figure 4 gives an overview of these values for the hard test set.

*Jaro Winkler:* The Jaro-Winkler approach gave good results on the mixed test set, but failed on the hard test set. The high F1-score of 0.80 on the mixed test set can be explained by the fact that many of the artificial aliases had a high Jaro-Winkler similarity. The hard test set more closely mimics a real-world scenario where aliases do not look as much alike. Therefore, the best F1-score achieved by Jaro-Winkler on this test set is only 0.28. However, the results still shows that using a simple string metric can detect many aliases resulting from spelling errors or the use of different email addresses for work, home, etc.

*Connected Path:* The Connected Path method achieves an F1-score of 0.48 on the mixed test set, and a score of 0.53 on the hard set. It can be concluded that the Connected-Path algorithm failed to achieve good results because of three reasons. First, since authors have been split up into aliases and some have been removed all together, the link network’s structure might have been corrupted. This especially affects link analysis that goes beyond the analysis of direct neighbors, since it takes into account more complicated link connections. Second, the link network search has been performed to depth 3, which means that only the information contained in paths of length 2 and 3 have been used in the calculation of the similarity score. Third, the Connected Path method can only return similarity scores for authors within close proximity of the original author. If there was no Connected Path score returned for a particular author-alias pair the alias had to be counted as a false negative.

*Jaccard:* Using Jaccard similarity yielded better results than the Connected Path algorithm: an F1-score of 0.69 and 0.67 for the mixed test set and hard test set respectively. Since the Jaccard similarity only takes into account direct neighbors, it is less affected by changes in the link network. Moreover, the Jaccard similarity can be calculated between any two authors in the data set, which is why it scored significantly better than the Connected Path method.

*Authorship SVM:* The use of authorship SVM’s gave good results overall, with an F1-score of 0.79 on the mixed test set and 0.76 on the hard test set. The results are especially good considering the fact that there are 314 candidate aliases for each author and that the training texts are short.

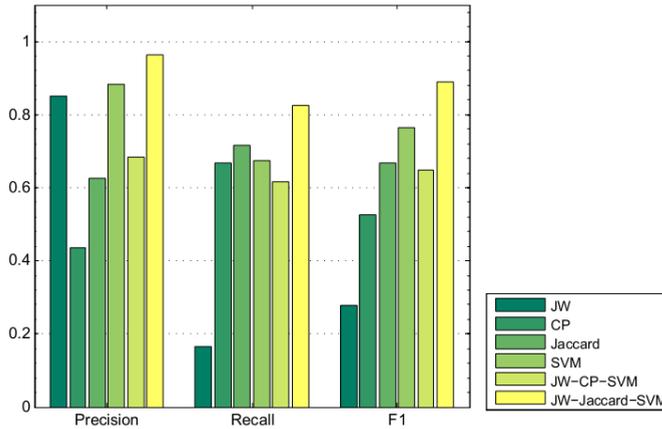


Figure 4: Precision, Recall and F1-scores for different techniques, evaluated on the hard test set.

*Combined techniques:* It can be concluded that the highest F1-score for both test sets is achieved by JW-Jaccard-SVM. For the mixed test set an F1-score of 0.88 was achieved, whereas on the hard test set an F1-score of 0.89 was achieved. These results confirm our hypothesis that a combination of techniques can yield better results than using these techniques individually. However, the combination of JW-CP-SVM on the mixed test set performed as good as authorship SVM or even Jaro-Winkler alone, with an F1-score of 0.80. For the hard test set it performed even worse, achieving an F1-score of 0.65. Because of aforementioned reasons, the Connected Path method failed to achieve good results in general. In combination with the low Jaro-Winkler performance on the hard data set this resulted in the combination JW-CP-SVM failing to achieve reasonable results.

## 4 Conclusion

The combination of Jaro-Winkler similarity, authorship SVM and Jaccard similarity outperforms individual and other combinations of techniques, achieving an F1-score of 0.89. It is important to note that the relative improvement in F1-score of the combined techniques over the individual techniques is dependent on the number of low Jaro-Winkler aliases in the test set. This indicates that the different techniques are indeed complementary and can work together to achieve better results. It can therefore be concluded from these results that it is beneficial to combine techniques from different domains using a voting SVM.

## 5 Future research

This paper showed that combinations of techniques can outperform the use of a single technique when applied to a real-life data set. It will be interesting to see how well these techniques perform on a full data set with real aliases, which could not be found to use in this research. Should such a collection not exist, it is worthwhile to create one.

The link analysis techniques that have been used in this paper only use information from the direct neighborhood of the authors. Boongoen et al.[1] have already shown that searching to a greater depth yields better results, so it would be useful to look at how the algorithm can be optimized to be less computationally intensive in order to search to greater depths.

Finally, the assumption has been made that the results from various techniques are independent of each other. These assumptions have not been tested, and it is not clear if and in what way various techniques affect each other. Therefore, it is important that more research will be done to examine the best choice of feature sets, techniques and aggregation methods.

## References

- [1] Tossapon Boongoen, Qiang Shen, and Chris Price. Disclosing false identity through hybrid link analysis. *Artificial Intelligence and Law*, 18(1):77–102, February 2010.
- [2] John Burrows. All the way through: Testing for authorship in different frequency strata. *Literary and Linguistic Computing*, 22(1):27–47, January 2007.
- [3] P. Christen. A comparison of personal name matching: Techniques and practical issues. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pages 290–294. IEEE, 2006.
- [4] William W. Cohen. Enron Email Dataset. Retrieved from: <http://www.cs.cmu.edu/~enron/>, 2009.
- [5] William W. Cohen, P. Ravikumar, and S.E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, pages 73–78, 2003.
- [6] O. Feiguina and G. Hirst. Authorship attribution for small texts: Literary and forensic experiments. In *Proceedings of the 30th International Conference of the Special Interest Group on Information Retrieval: Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection (SIGIR)*, 2007.
- [7] FERC. Information Released in Enron Investigation. Retrieved from: <http://www.ferc.gov/industries/electric/indus-act/wec/enron/info-release.asp>.
- [8] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.
- [9] C. Sanderson and S. Guenter. Short text authorship attribution via sequence kernels, markov chains and author unmasking: An investigation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 482–491. Association for Computational Linguistics, 2006.
- [10] Jitesh Shetty and Jafar Adibi. The enron email dataset: Database schema and brief statistical report. Technical report, Information Sciences Institute, 2004.
- [11] William E Winkler. The state of record linkage and current research problems. *Statistical Research Division US Census Bureau*, pages 1–15, 1999.
- [12] R. Zheng, J. Li, H. Chen, and Z. Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57(3):378–393, 2006.

# Depth-based detection using Haar-like features

Ruud Mattheij<sup>a</sup>   Eric Postma<sup>a</sup>   Yannick van den Hurk<sup>a</sup>   Pieter Spronck<sup>a</sup>

<sup>a</sup> *Tilburg center for Cognition and Communication (TiCC), Tilburg University, P.O. Box 90153, 5000 LE, Tilburg, The Netherlands, R.J.H.Mattheij@uvt.nl*

## Abstract

The automatic detection of objects has gained considerable attention over the last few years. Most object-detection approaches rely on visual features that are sensitive to identity-irrelevant variations, such as changes in illumination. Being less sensitive to such variations, depth features may improve detection accuracy. Depth features can be extracted from depth images generated by commercially available depth sensors, such as Microsoft's Kinect device. This paper describes a method for robust and accurate face detection by employing Haar-like region features on the integral image representation of depth images. Our aim is to determine to what extent region-comparison features contribute to effective face detection in depth images, compared to pixel-comparison features. To this end, we present a revision of the recently proposed detector of Shotton et al. [10]. Whereas the detector of Shotton et al. relies on pair-wise pixel comparisons in depth images, our revision compares square regions in a pair-wise fashion. In a comparative evaluation of the original and revised method, we train and evaluate both detectors on our depth images of faces (DIOF) database that is compiled at our lab. The results reveal that the use of region features instead of pixel pairs indeed improves face detection accuracy in depth images. We conclude that employing region features contributes significantly to effective face detection. Future work will address to what extent our results generalize to the detection of body parts and objects in general.

## 1 Introduction

During the last few years, the automatic detection of objects, such as human body parts, from digital video and image sources has gained considerable attention within the field of image analysis and understanding [7, 12]. Many approaches towards object detection focus on feature-based detection [4, 6]. A well-known example of feature-based detection is the use of Haar-like rectangle features in the state-of-the-art face detector proposed by Viola and Jones [11]. In the Viola-Jones face detector, the rectangle features enable efficient and fast face detection. Despite this success, the detector is sensitive to changes in illumination [14, 15]. Using additional or alternative cues such as depth information [9] may help to overcome such sensitivities by providing illumination-invariant cues, which can potentially make object detectors more robust [2]. Employing depth cues is made possible by commercially available depth sensors like the Microsoft Kinect device.

Shotton et al. [10] proposed a depth-based detector that is able to quickly and accurately classify body joints and parts from single depth images. Their method employs depth-comparison features defined as pixel pairs in depth images. The use of pixel-based features makes their method computationally efficient which allows for real-time operation. The computational efficiency comes at the cost of noise sensitivity. Averaging over larger regions of the depth image reduces the noise and may lead to an improved accuracy.

In this study, we use depth cues for robust and accurate face detection in depth images. Inspired by the work of Viola and Jones [11], we propose a revision of the recently proposed detector of Shotton et al. [10]. Whereas the detector of Shotton et al. relies on pair-wise pixel comparisons in depth images, our revision employs Haar-like features by comparing square regions in a pair-wise fashion. Although for visual images, the use of region features introduces illumination sensitivity, for depth images they may improve upon the noise-sensitivity of pixel pairs. The aim of our study is to determine to what extent region features contribute to effective face detection in depth images. To achieve this aim, we perform a comparative evaluation to determine whether our region-comparison detector yields an improvement with respect to the original pixel-comparison detector.

## 1.1 Related work

Our region-comparison detector is related to two recent methods for object detection in depth images. The first related method was proposed by Xia et al. [13]. Their method detects head shapes by means of a generic model of the 2D contour and the 3D depth map of the head. The generic model is detected by means of fast convolution. Our revised method differs from Xia et al.’s method in the use of local features, instead of a global shape model. The second related method is due to Plagemann et al. [8] who proposed a method to detect and identify body parts in depth images. Their method identifies points of interest that are based on the differences in geodesic distances, which coincide with salient points of the body. In their method, the shape of the surface meshes is defined by points with similar geodesic distances. A commonality between their method and our detector is that local depth information is used. The main difference is that we apply a series of pre-defined feature types instead of attempting to identify points of interest.

## 1.2 Outline

The outline of the remainder of the paper is as follows. Section 2 reviews the pixel-comparison detector of Shotton et al. [10] and presents our region-comparison detector. Section 3 describes the experimental methodology used to determine the accuracy of our detector and presents the results of the evaluation. Section 4 discusses results and we conclude on our findings in Section 5.

# 2 Pixel comparison versus region comparison

In this section, we describe the detector proposed by Shotton et al., henceforth referred to as the pixel-comparison detector (subsection 3.1), and present our revision, the region-comparison detector (3.2). Finally, we outline the randomized decision forest classifier that was trained using the region features (3.3).

## 2.1 The pixel-comparison detector

The pixel-comparison detector of Shotton et al. [10] employs simple and computationally efficient depth-comparison features to identify different skeletal joints and body parts. Figure 1a shows an overview of the detector. To calculate the features, a subset of random pixel positions is selected from each depth image. (The subset is different for each depth image.) For each position P from this subset, the feature value is computed by comparing the depth value at two offset locations Q and R. The offset locations are defined by the radius and angle with respect to P. The radius is defined to be inversely proportional to the depth value of P. A small depth value results in a larger radius for offset positions P and Q, and vice versa. In this way, a scale-invariant measure of depth is obtained. In our implementation, the angles are defined to be multiples of  $30^\circ$  and selected through exhaustive search as to obtain those angles that give the largest difference in the depth values of Q and R. Although an effective approach for feature selection, this might influence the prediction time. Figure 2 shows two example features. The use of pixel pairs as basic features makes this method fast and computationally very efficient at the cost of errors introduced by the use of individual pixel values which may be noisy.

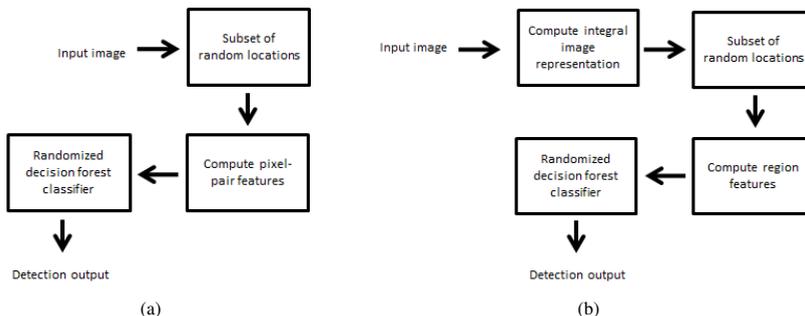


Figure 1: An overview of (a) the pixel-comparison detector that employs pixel pairs as depth features, and (b) our region-comparison detector that employs square regions in a pair-wise fashion, i.e., region features.

For each depth image, the pixel-comparison detector yields a vector with depth features that provide a probabilistic cue about the part of the body sampled. The feature vectors provide the inputs to a randomized decision forest [1] for classification.

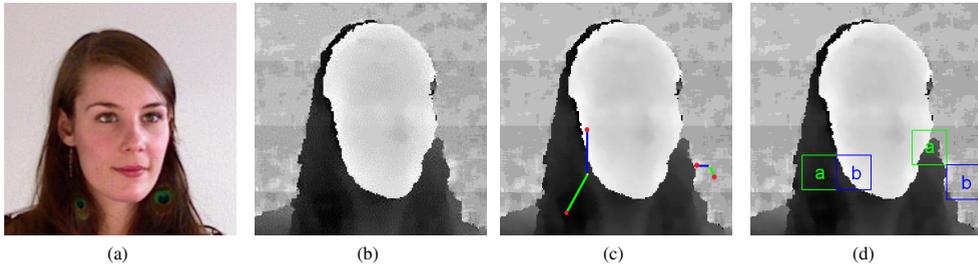


Figure 2: (a) Example of a visual image from our depth images of faces (DIOF) database (described in section 3.1), and (b) the corresponding depth image. (c) Illustration of two pixel-based depth-comparison features, and (d) two Haar-like region features.

## 2.2 The region-comparison detector

The detector is an improvement of the pixel-comparison detector proposed by Shotton et al. [10] and employs two of the contributions proposed by Viola and Jones [11]: (1) the Haar-like region features, and (2) the integral image representation. Below, we will briefly address these contributions and describe how they are employed to improve the pixel-comparison detector. Figure 1b shows an overview of our detector. As in the pixel-comparison detector, a subset of random pixel positions is selected from each depth image. Region features are two-dimensional filters (or masks) that respond to vertical, horizontal, or diagonal contours and bars in an image. They are based on the well-known Haar wavelets [5]. The features are defined in terms of square regions in an image, hence their name. A region feature  $f$  for position  $P$  (of one of the pixels in the random subset) in depth image  $I$  can be computed by calculating the sums  $S$  of the pixels enclosed by two square areas and subtracting these sums from each other. This results in a single feature value  $f(I, P)$ . This feature value provides an indication of the direction and magnitude of the depth transition over an area in a depth image.

In what follows, we describe the computation of the feature values in more detail. Feature values depend on (1) the parameter  $r^2$  defining the size of the individual square regions, and (2) the configuration  $i$  defining the orientation of the constituent square regions of the feature.

The sizes of the square regions define the area over which the depth difference is calculated. Employing larger squares for the region features results in a feature value that describes the depth transition over a larger area in the depth image. By calculating the sum of the square areas for all possible square sizes  $r^2$  (which can be achieved very efficiently using the integral image), we ensure that the region features capture a large range of head sizes.

The feature type describes the locations of the two constituent square regions in relation to each other, thereby providing an indication of the direction of the depth transition. Figure 3 illustrates the four pairs of feature types that we employed in the region-comparison detector, which allow for the detection of horizontal, vertical, diagonal and anti-diagonal depth transitions. In the figure, the green square represents region  $S_i(x_a, y_a, r^2)$  and the blue square represents region  $S_i(x_b, y_b, r^2)$ . For all possible combinations of  $r^2$  and  $i$  on position  $P$ , we compute the feature value  $f(I, P)$  as proposed by [11]. We then select the highest value as the feature value for that position. The addition over feature types is performed to Although region features are less sensitive to erroneous pixel values because they average over many pixels, these features may require more computation than the pixel-pair features employed by the pixel-comparison detector. Fortunately, region features can be computed rapidly using an alternative image representation called the integral image representation [11]. Adopting the integral image representation for depth images allows for a considerable speed-up.

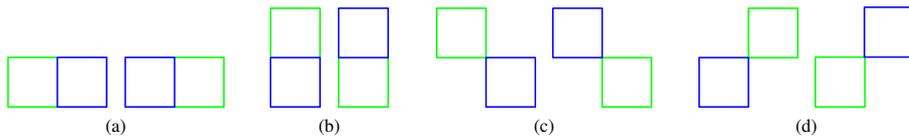


Figure 3: The feature types used in the region-comparison detector: (a) horizontal features, (b) vertical features, (c) diagonal, and (d) anti-diagonal region features.

### 2.3 The randomized decision forest

The random selection of  $N$  pixel positions in an image, results in an  $N$ -dimensional feature vector. A randomized decision forest classifier [3] is used to perform the binary classification on the basis of these vectors. Randomized decision forests are fast and effective classifiers that employ an ensemble of decision trees for prediction. Each individual tree consists of binary split- and leaf nodes. Individual split nodes compare single features from the feature vector with a threshold, branching left or right depending on the outcome of the comparison. The leaf nodes of each tree contains the prediction result. The predictions of all decision trees are then averaged over the ensemble of trees, thereby giving the final classification.

## 3 Experiment and Results

In this section we describe the database that was used to train and evaluate our detector (3.1), and the comparative evaluation of the pixel-comparison and region-comparison detector (3.2). We train both detectors on a database with depth images of faces that we compiled in our lab.

### 3.1 The Depth Images Of Faces database

The database used to train and evaluate the detectors was the depth images of faces (DIOF) database that was compiled at our lab. It contained visual and depth images of human faces under various lighting conditions and distances. Figure 2 shows an example of a visual image and the corresponding depth image of a participant's face. The database assembled images of 100 participants (51 male and 49 female). We employed a Microsoft Kinect device to create visual images with a resolution of  $1280 \times 1024$  pixels and depth images with a resolution of  $640 \times 480$  pixels. For every participant, a series of depth images were created on five distances from the Kinect device: 0.5 meters to 2.5 meters, with steps of 0.5 meters. For each distance from the Kinect device, we created depth images under five distinct lighting conditions: dim light, environmental light, fluorescent ceiling light, intense frontal light and intense light from the left side of the participant.

A face detection algorithm was applied to annotate the location of the participants' face in the visual images. The region was selected in the corresponding depth image and labeled as a positive (containing a face) example. Likewise, negative examples were selected by labeling non-annotated regions from the depth images. Given the various distances at which the images were taken, the dimensions of the example depth images varied between  $75 \times 75$  pixels and  $450 \times 450$  pixels. For the final database, we randomly selected 1000 positive and 1000 negative example depth images.

### 3.2 Evaluating the region-comparison detector

The aim of our experiment is to investigate to what extent region features contribute to effective face detection in depth images as compared to pixel features. We address this aim by training and evaluating the pixel-comparison detector and region-comparison detector on our DIOF database. We repeat the experiment while employing feature subsets of various sizes, starting with a subset of 1 feature per image, up to a subset of 2000 features per image, with steps of 5 features per image.

For the comparative evaluation we employ 10-fold cross-validation. For every fold, the evaluation of both detectors is performed on the same set of training and test images. Each fold consists of 1800 training examples (900 positive and 900 negative examples) and 200 test images (100 positive and 100 negative examples). The training examples are used to train a randomized decision forest consisting of 50 trees, while the test images are used to evaluate the performance of the detectors. For both detectors,

the average detection performance over all folds is shown in table 1 (the pixel-comparison detector) and table 2 (our region-comparison detector). Both tables report the *Accuracy* as a performance measure, defined as  $(TP + TN)/(TP + FP + FP + TN)$  where  $TP$  represents the number of true positives,  $FP$  false positives,  $TN$  true negatives, and  $FP$  false positives. In addition, *Recall* ( $TP/(TP + FN)$ ), and *Precision* ( $TP/(TP + FP)$ ), are reported. All measures are expressed in percentages. The entire training- and evaluation sequence took approximately 48 hours on a 24-core Linux computation server.

Number of features	Accuracy (%)	Recall (%)	Precision (%)	Prediction time (s)
1	64.0 (2.29)	55.8 (5.14)	66.7 (2.19)	0.81 (0.04)
5	69.2 (3.74)	71.6 (4.45)	68.3 (3.69)	1.23 (0.06)
15	72.9 (2.29)	82.8 (3.29)	69.1 (2.00)	1.76 (0.09)
25	75.6 (2.71)	86.7 (3.06)	70.9 (2.57)	2.17 (0.15)
2000	77.0 (2.79)	87.7 (3.06)	72.3 (3.03)	105.3 (3.94)

Table 1: Average detection performance on subsets of various sizes, expressed in percentages (accuracy, recall, and precision) or seconds (prediction time) and, between brackets, the corresponding standard deviations for the pixel-comparison detector.

Number of features	Accuracy (%)	Recall (%)	Precision (%)	Prediction time (s)
1	72.4 (2.93)	72.2 (4.34)	72.4 (2.73)	0.84 (0.02)
5	86.0 (1.91)	87.5 (3.06)	84.9 (2.15)	0.90 (0.05)
15	87.5 (2.15)	89.0 (3.37)	86.4 (2.01)	0.92 (0.05)
25	90.3 (3.02)	88.1 (2.06)	90.3 (3.02)	0.88 (0.03)
2000	88.5 (1.80)	90.4 (2.76)	87.1 (1.81)	5.98 (0.27)

Table 2: Average detection performance on subsets of various sizes, expressed in percentages (accuracy, recall, and precision) or seconds (prediction time) and, between brackets, the corresponding standard deviations for our region-comparison detector.

The results of the experiment indicate that the region-comparison detector achieves a significantly higher detection accuracy and precision than the pixel-comparison detector. The recall is slightly better. The results also indicate that the region-comparison detector achieves a considerably shorter prediction time and therefore a higher prediction speed than the original method.

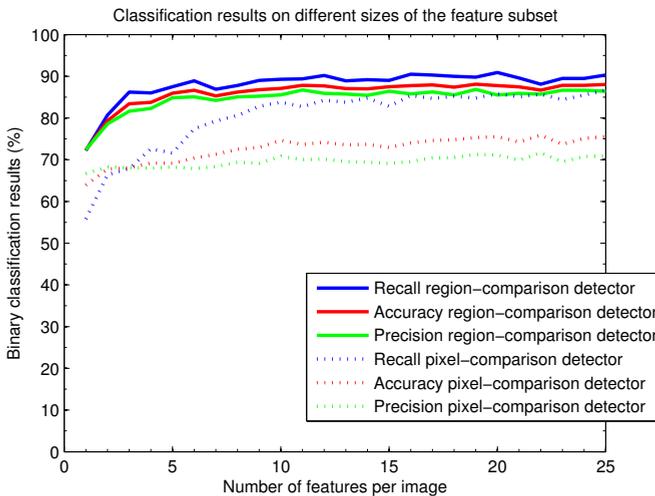


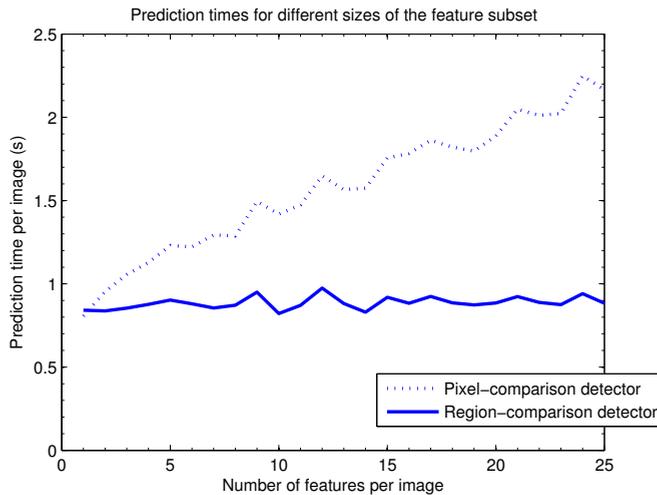
Figure 4: Average detection performance (accuracy, recall and precision, expressed in percentages) for the pixel-comparison detector (dotted line) and the region-comparison detector (solid line) on subsets of various sizes.

Our experiments suggest that employing a small feature subset may already achieve a high detection performance. Figure 5 shows the average detection performance (accuracy, recall and precision) for the pixel-comparison detector and our region-comparison detector for subsets of various sizes. Our experiments indicate that the optimal detection performance on our database is achieved while employing 25 features per image.

For a subset of 25 features per images, the region-comparison detector yields an average accuracy of 90.3% ( $\sigma = 3.02\%$ ), with an average recall and precision of 88.1% ( $\sigma = 2.06\%$ ) and 90.3% ( $\sigma = 3.02\%$ ). The pixel-comparison detector method achieves an average accuracy of 75.6% ( $\sigma = 2.71\%$ ), with an average recall and precision of 86.7% ( $\sigma = 3.06\%$ ) and 70.9% ( $\sigma = 2.57\%$ ), respectively. The prediction time for the region-comparison detector is 0.88 seconds ( $\sigma = 0.27$  seconds) per image, while the prediction time for the pixel-comparison detector is 2.17 seconds ( $\sigma = 0.15$  seconds) per image. For this number of features per image, the region-comparison detector yields a significantly higher detection accuracy and precision than the pixel-comparison detector while the region-comparison detector is approximately 2.5 faster than the pixel-comparison detector.

A highly similar pattern of performance results is obtained after employing a subset of 2000 features per image. The pixel-comparison detector achieves an average accuracy of 77.0% ( $\sigma = 2.79\%$ ), with an average recall and precision of 87.7% ( $\sigma = 3.06\%$ ) and 72.3% ( $\sigma = 3.03\%$ ), respectively. The region-comparison detector achieves an average accuracy of 88.5% ( $\sigma = 1.80\%$ ), with an average recall and precision of 90.4% ( $\sigma = 2.76\%$ ) and 87.1% ( $\sigma = 1.81\%$ ). The prediction time for the pixel-comparison detector is 105.3 seconds ( $\sigma = 2.76$  seconds) per image. The prediction time for the region-comparison detector is 5.98 seconds ( $\sigma = 0.27$  seconds) per image. For this subset, the region-comparison detector yields a significantly higher detection accuracy and precision than the pixel-comparison detector, while performing approximately 17 times faster than the pixel-comparison method.

Figure 5 shows the prediction times for both detectors on various subset sizes. The results indicate that the prediction time for the pixel-comparison detector increases significantly faster than the prediction time of the region-comparison detector.



(a)

Figure 5: Average prediction times (expressed in seconds) for the pixel-comparison detector (dotted line) and the region-comparison detector (solid line) on subsets of various sizes.

The results of the comparative evaluation of the region-comparison detector on the DIOF database show that the combination of region features and the integral image representation allows for fast and effective face detection in depth images. By employing region features, the region-comparison detector achieves a significantly higher detection accuracy and precision than the pixel-comparison detector.

## 4 Discussion

The results of our comparative evaluation show that the region-comparison detector achieves a high detection accuracy and precision than the pixel-comparison detector. Below, we briefly discuss three points regarding our findings: the validity of the experiment (4.1), the effect of rectangle features (4.2), and the points of improvement for our face detector (4.3).

### 4.1 Validity of the experiment

The results of the evaluation suggest that employing region features in combination with the integral image representation improves the detection accuracy of faces in depth images significantly, while maintaining a short prediction time. These results are achieved by training and evaluating the pixel-comparison detector and the region-comparison detector on the DIOF database for a range of feature subset sizes. As no implementation of the pixel-comparison detector was available, we developed our own implementation of the detector proposed by Shotton et al. [10].

Our choice of performing an exhaustive search for selecting appropriate pixel pairs (see section 2.1) has the advantage that it yields the largest difference in depth values, but may impose a computational cost. In future work, we should examine the effect of various realization of the selection algorithm on the performance and speed of the pixel-comparison detector.

The evaluation of the region-comparison detector was performed on the DIOF database. Although the results suggest that employing region features improves the detection accuracy and precision, our results are limited to the task of frontal face detection. Depth images of faces with other orientations or other body parts are not present as distinguishable classes in our database. Future versions of our experiment should adopt a more challenging database with depth images with a larger variety of face orientations and other body parts.

### 4.2 Effect of region features

The results of our evaluation show that employing region features for detection tasks in depth images results in fast and accurate face detection. The pixel-comparison detector experienced difficulties calculating feature values in noisy depth images. For example, instances in which parts of the participants' faces were too close or too far from the depth sensor of the Kinect device tended to result in erroneous pixel values. Apparently, as the region features employed in our region-comparison detector average over many pixels, these features are less sensitive to background noise in depth images, compared to the pixel-pair features employed by the pixel-comparison detector.

### 4.3 Points of improvement

We identify three main improvements of our detector and the database that is used for the detector's training and evaluation procedure.

First, our implementation of the detector of Shotton et al. [10] should be validated by applying it to one or more of the data sets employed in their original paper. In this way, we will be able to assess the validity of our implementation.

Second, the region-comparison detector might be improved by employing it for detection tasks that involve faces and other body parts in depth images with other orientations and poses.

Third, the region-comparison detector classifies entire images as containing a face or not a face. The region-comparison detector should be extended by enabling it to investigate points of interest in a given depth image, so it can locate specific areas in an entire depth image that might contain heads.

## 5 Conclusion

The aim of our study was to determine to what extent the combination of region features contribute to fast and effective face detection in depth images. To achieve this aim, we proposed the region-comparison detector that employs Haar-like region features on the integral image representation of depth images. We trained and evaluated the region-comparison detector and the pixel-comparison detector proposed by Shotton et al. on the depth images of faces (DIOF) database.

The evaluation of the region-comparison detector revealed that employing region features results in fast and accurate face detection in depth images. The results show that the detector yields a significantly higher accuracy and precision than the pixel-comparison detector. Combining region features with the integral image representation results in a short prediction time. The results indicate that a large subset of features per image does not necessarily lead to better detection results. The experiments indicate that a significantly smaller feature subset can also yield a high detection performance.

We conclude that employing region features contribute significantly to fast and effective face detection in depth images and that the region comparison-detector yields an improvement over the detection accuracy of the pixel-comparison detector.

## References

- [1] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [2] W. Burgin, C. Pantofaru, and W. D. Smart. Using depth information to improve face detection. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 119–120, New York, NY, USA, 2011. ACM.
- [3] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3):81–227, February 2012.
- [4] G. C. H. E. De Croon, E. O. Postma, and H. J. Van Den Herik. Adaptive gaze control for object detection. *Cognitive computation*, 3(1):264–278, 2011.
- [5] J.. Guf and W. Jiang. The haar wavelets operational matrix of integration. *International Journal of Systems Science*, 27(7):623–628, 1996.
- [6] M.W. Lee and R. Nevatia. Body part detection for human pose estimation and tracking. In *Proceedings of the IEEE Workshop on Motion and Video Computing, WMVC '07*, pages 23–, 2007.
- [7] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In Tomáš Pajdla and Jiri Matas, editors, *European Conference on Computer Vision (ECCV '04)*, volume I, pages 69–82, Prague, Czech Republic, 2004. Springer-Verlag.
- [8] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3108–3113, may 2010.
- [9] N. Riche, M. Mancas, B. Gosselin, and T. Dutoit. 3d saliency for abnormal motion selection: The role of the depth map. In J. Crowley, B. Draper, and M. Thonnat, editors, *Computer Vision Systems*, volume 6962 of *Lecture Notes in Computer Science*, pages 143–152. Springer Berlin / Heidelberg, 2011.
- [10] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. *CVPR*, 2:3, 2011.
- [11] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition (CVPR)*, 1:511–518, 2001.
- [12] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *Int. J. Comput. Vision*, 75(2):247–266, November 2007.
- [13] L. Xia, C.C. Chen, and J.K. Aggarwal. Human detection using depth information by kinect. *Workshop on Human Activity Understanding from 3D Data in Conjunction with CVPR (HAU3D)*, pages 15–22, 2011.
- [14] C. Zhang and Z. Zhang. A survey of recent advances in face detection. *Learning*, (June), 2010.
- [15] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35:399–458, 2003.

# Towards a Universal Change Detection Framework in Levees

Gabriel Mititelu<sup>a</sup> Robert-Jan Sips<sup>b</sup> Bram Havers<sup>b</sup> Zoltán Szlávik<sup>a</sup>

<sup>a</sup> Dept. of Computer Science, VU University Amsterdam, 1081 HV Amsterdam

<sup>b</sup> Center for Advanced Studies, IBM Nederland B.V., 1066 VH Amsterdam

## Abstract

Floods are an increasing threat to civilisation, they are among the most destructive of all natural disasters. With more than half of the world's population living along coastlines, lakes and rivers, and with important economical activities located in such areas, the potential damage from flooding is enormous. The most important defence against floods are levees. But levees can break, and they need to be actively monitored. Recently, modern ways of monitoring have been implemented, with the use of sensor systems. Via these systems, information about pore pressure, humidity, inclination, movement and temperature, etc. are being recorded. The data gathered is used for modelling the behaviour of the levee, and to assess its stability and resistance. It can also provide insights into how a levee will behave in the future, by signalling critical events upfront. However, currently there is no generally accepted way to predict such events using the recorded data. Traditionally, forecasting of levee strength involves the use of specialist domain knowledge on, e.g., the ground layers of which the levee is made up. The trend of using more data generated by sensors provides new possibilities for using statistical methods to forecast levee strength – and potential failure – that use less domain knowledge and are therefore more generally applicable.

In this paper we describe framework to detect changes in levee behaviour. Our approach uses clustering as a primary step. After obtaining clusters, different indicator measures are computed, such as number of active clusters at a certain point in time, distances among clusters, etc. The set of these indicators serve as the basis for several change identification methods applied. These are compared and the advantages and downsides for each them are identified and discussed.

Our results show that, by using the proposed framework, changes can be detected in levee sensor data effectively.

## 1 Introduction

Current flood protection is primarily concerned with strong levees. More than half of the world's population lives along coastlines, lakes and rivers; the deltas are densely populated and are becoming economically more valuable [1, 2]. The work presented in this paper was carried out in such a critical area of the world, the Netherlands, where 55% of the land is prone to flooding from the sea and rivers, which makes levee management critical.

The currently followed approach to safe design of levees in the Netherlands is based on the findings of the Delta Committee that were published in 1960 [3]. The design principles are mainly founded on the likelihood of exceeding the *design water level*, combined with a *return period*. The former describes the highest water level that a levee is designed for, while the latter is the period in which this is expected to occur, e.g. the standard for Central Holland is 5m above NAP in every 10,000 years [3]. NAP is the Amsterdam Ordnance Datum – a reference for measuring water levels.

The total length of levees in The Netherlands is around 21.000 km, of which ~4500 km are primary dikes and ~18000 km are secondary (regional) levees. Every 6 years these levees are inspected for meeting the safety standards. The inspection report of 2011 [4] shows that 32% of the dikes and dunes are

not up to design safety norms, and that 6% needs additional inspections, which shows that more effective ways to deal with levee quality assessment, and also quality prediction, are needed.

Despite the long history in levee building and their high importance for –especially– the Netherlands, surprisingly little is known about their internal behaviour [5]. Although the failure models that can impact levee integrity are rather easy to understand, predicting and modelling them proves to be challenging. Our current geophysical models seem insufficient to capture both the scale [5] and the actual phenomena [2, 5] found in real levees. The recent levee-failure in Wilnis could only be explained by a phenomenon unknown to that date [5, 6].

Due to the lack of geophysical knowledge and the availability of new datasets arising from recent investments [7, 8, 9] in sensor-monitoring, machine learning may be a feasible approach to detect the need for maintenance and to predict possible events in levees, agnostic of the geophysical structure and model underlying these events.

In this paper, we focus on detecting and predicting change in levees, using sensor data from three distinct levee experiments. One of the main challenges that we are addressing is that, although considerable amount of data is available about levees, there is (thankfully) not a lot of data available about events where levees failed. This makes it impractical to consider any supervised modelling method, and so, other ways to describe and predict levee behaviour need to be investigated. In this paper, we describe a framework that is primarily based on clustering.

## 2 Data from Three Levee Experiments

To develop a framework for change-detection in levees, we acquired three datasets: IJkDijk, LiveDijk [9] and LekDijk. Using these three inherently different datasets allowed us to test our approach on different data in terms of 1) robustness of the proposed framework with regards to data-collection, and 2) type of levee and –consequently– change-events that may be present in the data, e.g. a sea- levee may be subject to tidal influences, which will typically not impact a river-levee, thus testing the robustness with regards to change-types is an important issue. Table 1 provides a summary of the datasets used in our experiments.

Data source	Type of levee	Records	Attributes	Reading interval
IJkDijk	Artificial	5074	378	60s
LiveDijk	Real (sea)	30789	375	300s
LekDijk	Real (river)	8822	48	Variable

**Table 1: Data Summary**

Our first dataset comes from the **IJkDijk** experiment. The IJkDijk was an artificial levee, which was made to collapse in an experiment on the 27th of September 2008. The purpose of the IJkDijk experiment was twofold: First, it was intended to test different sensor-systems that may be used to monitor levees in practice, and second, it involved monitoring the events occurring in an (artificially induced) levee failure. The experiment ran for 3.5 days, until the levee collapsed. The experiment was labelled successful, as the levee, built specifically for the experiment, collapsed as intended and the sensors gathered useful data. From this experiment, we obtained a dataset of 5074 data points, describing pressure, humidity, inclination, movement and temperature at 60 second intervals.

The **LiveDijk** is the world's first sea levee equipped with sensors that allow remote monitoring of the status of the levee. It is the previously tested setup in the IJkDijk experiment taken into a real sea levee. The levee is not part of the direct sea barrier but it has contact with the sea. This means that the risk at this levee is reduced, but it offers a good testing environment for the sensors. The corresponding data is without extreme events, great or sudden changes, and so, it offers a good data set to test sensor behaviour and the change detection framework. Here we expect to identify small changes caused by tidal influence. The data covers a period of 8 months. For (approximately) every 5 minutes a reading is stored in the dataset.

The third dataset comes from the **LekDijk** levee, a real river levee equipped with sensors. The period of time captured in the data analysed is from 16/04/2010 to 01/03/2011. The parameters measured are the hydraulic level, inclination, pressure inside the levee, and also air pressure. Measurements are taken at variable time intervals, differing from parameter to parameter. For example, inclination is measured every hour, hydraulic level and pressure every 6 hours.

### 3 Change/Outlier Detection

In recent years, various authors have stressed the need for automated monitoring of flood-defence structures [7, 8, 10, 11, 12]. Most research, including the research that resulted in the IJkDijk and LiveDijk datasets [9, 12] focussed primarily on testing the feasibility of sensor-monitoring and on assessing the quality of different types of sensors and setups.

However, several authors, particularly Naruse *et al.* [7], Pyayt *et al.* [11] and Krzhizhanovskaya *et al.* [10] have stressed the importance of automated *change-detection* on these sensor-monitored structures. Corresponding experiments were conducted either on a single sensor-type [7], a single levee [11], or a simulated model [10]; therefore not guaranteeing applicability on both real-life scale as well as real-life events. Our goal is to carry out change detection by addressing at least some of these shortcomings.

Changes/events can be defined as time points at which the properties of time-series data change [13]. The process of identifying such points is known in literature as change-point detection [16, 17] or as event detection [16]. Recently, there has been increased interest in using data mining techniques for extracting interesting patterns from time series [17], and for discovering such change points [18].

Change detection has always been presented in strong relation with outlier detection [19]. Indeed, a change occurring in time can be seen as an outlier. The difference between the traditional view on what an outlier is, and what changes or events are is the fact that the word 'outlier' is also used in contexts in which the time dimension is not present.

In this paper, we define our framework (see Section 4) using the outlier detection point of view. To place our approach in context, we consider the survey paper of Chandola *et al.* [20], who distinguish various kinds of outliers, as well as different classes of outlier detection methods. In case of the levee data we have available, we are looking for 'contextual' and/or 'collective' outliers. While choosing the right approach in solving the problem we considered a number of method classes, e.g., classification based outlier detection, nearest neighbour based outlier detection, clustering based outlier detection, etc. The approach we have chosen is based on clustering. It has a natural way of coping with the fact that data labels often lack. The following section describes a change detection framework that is based on clustering, after which we present and discuss results of our experiments on the three datasets used.

## 4 Clustering-based Change Detection

Our framework consists of three main steps. In the first step, we propose to **create clusters** of the data. Then, based on the output of the clustering, **indicator measurements are computed**, which will indicate at a particular timestamp if changes occur and they serve as a basis for the third step, where **changes are identified** by analysing the values of the indicator measures. One of the main ideas here is that if there are several clusters starting or ending at a certain point in time, then it is likely that an event/change is happening at that time point.

The framework is developed and trained on recorded data but the aim is to make it usable in an online setting. This will be made available by comparing new points with the existing created clusters. In order to keep the model to reflect the normal situation, re-clustering is to be carried out regularly.

### 4.1 Clustering

As a first step towards change detection in levees, we propose to use clustering. Our aim here is to cluster measurements at different points in time, and see whether (and if so, how many) clusters will actually cover continuous time periods. Note that time information is not used for clustering, but we expect that clusters will be associated with time periods. For example, one cluster may contain low tide data, another rising tide data, etc.

Because the number of clusters we can effectively use is unknown, we chose a method that does not require this as parameter that heavily influences the outcome. For our experiment, we chose Kohonen's self-organising map algorithm for clustering [21]. Kohonen clustering is applied on the preprocessed data. An additional advantage of this method over other clustering methods (e.g. K-means) is that the resulted clusters are also mapped on a 2D axis system according to how related clusters are to one another, and that it is an on-line algorithm (inputs are presented one by one). The size of the 2D axis system we used at the input is chosen to be large enough to provide sufficient cluster granularity.

As mentioned above, the clustering step ignores any time- or date-related attributes. Our purpose is to try to detect similar behaviour with no time reference.

### 4.2 Computing Indicator Measures

After generating clusters, we need to find a way to identify when changes occur. The identification is done by using so called indicator measures. Indicator measures are built by using the properties of the clusters computed in the first step. For all indicators we project the clusters on the time dimension.

As part of this step of the framework, we computed three types of indicators (though others might also be used):

a) **Number of boundaries.** Given a window size, in every time interval with the size of this window we can calculate the number of clusters starting or ending within this period. We say that a window contains a cluster boundary if the point with the smallest/largest time stamp of a cluster is contained within the time period corresponding to the window. The intuition behind this measure is that changes/events are indicated by multiple (often small) changes, which are reflected through multiple clusters starting/ending. Our experiments (described in the next section) indicate that the framework as a whole is quite robust to the choice of window size (typically being around 1-5 hours in width).

b) **Number of active clusters.** Using a similar intuition to above, we can calculate the number of 'active' clusters within a window, i.e. the number of clusters that have corresponding points within a window.

c) **Distances between clusters.** The third group of indicator measures uses the distance between active clusters on the 2D surface generated by Kohonen clustering. Whenever a new cluster becomes 'active' as time passes (leftmost points in each row in Figure 1), the value of the indicator measure changes the following way: We take the newly active cluster, and calculate its distance to other active clusters. The following indicator values are calculated: minimum, maximum or mean distance of the new cluster to the others, either using Euclidean or Manhattan distance. This gives us six distance based indicator measures that might indicate the status of the levee investigated.

### 4.3 Identifying Changes

Up to this step we have computed various measures that should reflect changes in levee behaviour. The third step of the framework aims to automatically pinpoint the moments in time when changes occur. We propose three alternative ways to do this, though other methods could also be used as well. In order to identify changes we consider the following:

a) If a **change in indicator value** from one point in time to another is above a threshold (determined using the labelled IJkDijk data), we signal an event.

b) **K-means clustering**, where we –again– remove time information, and cluster values of the indicators. The rationale here is that we can identify various levels of events (number of levels determined by the value of  $k$ ) by looking at clusters and their sizes. For example,  $k=3$  would allow us to distinguish between normal behaviour (expected to correspond to the largest cluster), and minor as well as major events (smaller and smallest clusters, respectively).

c) **Kohonen clustering**, where the main idea is the same as for b) above, but this time the number of levels is also automatically determined.

The next section presents and briefly discusses experimental results when applying the above framework on the IJkDijk, LiveDijk and LekDijk data.

## 5 Results

The framework described above has been applied in the context of our three datasets. In this section, we present the results of our experiments.

**IJkDijk** The experiments carried out using the IJkDijk data aimed to capture various –known– phases of the process at the end of which the levee collapsed. These events are marked using vertical red lines in Figure 1, a bold line indicating when the levee collapsed. Figure 1 also shows cluster allocations over time. It is worth noting that, although our clustering step did not take into account time, points of many clusters tend to be closely aligned in time. We can also observe that the process at the end of which the levee collapses is gradual, which is indicated by the presence of close clusters, also over time. Once this gradual change (the build-up phase) ends, i.e. when the levee collapses, we can see several new clusters 'starting'. In short, gradual change might *predict* a larger event, while many new clusters present *indicate* that a significant event is already happening.

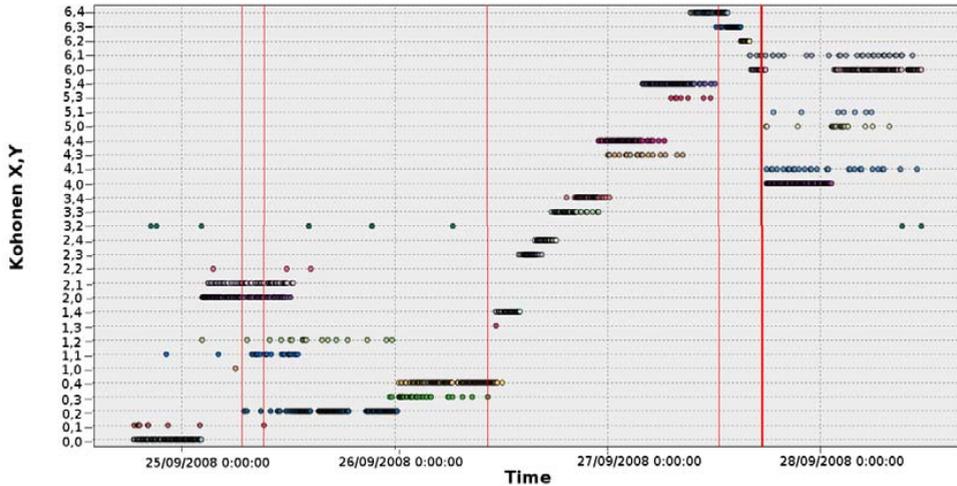


Figure 1: Clusters plotted over time for the IJkDijk data. Points in each row correspond to individual clusters, red lines are significant events in levee behaviour.

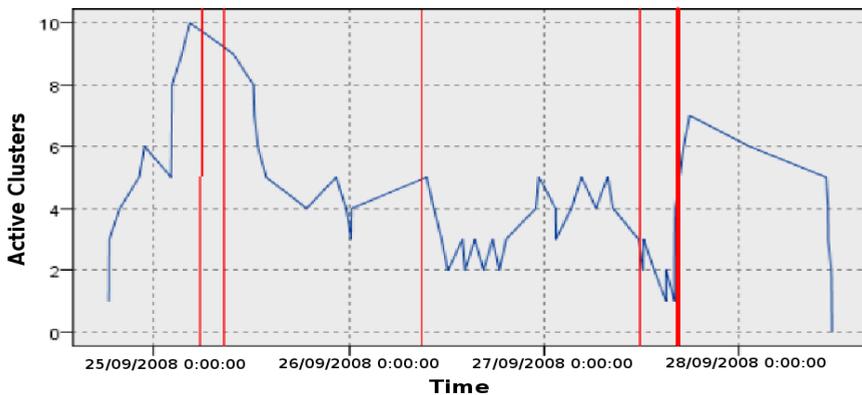


Figure 2: Number of active clusters, an indicator calculated from the IJkDijk data.

Figure 2 shows how the values of one particular indicator measure, i.e. the number of active clusters, change over time. Step 3 of the framework aims to identify changes in the levee's behaviour using these values. Using the IJkDijk data, we were able to identify the major event and all the small events by using the number of active clusters as indicator, and k-means clustering ( $k=3$ ) in the final step. If using cluster distances in Step 2, a lower recall was achieved, but all identified events were labelled important by experts. These results show that the framework is indeed capable of identifying various changes in levees.

**LiveDijk** Applying the framework on the LiveDijk data provided similar conclusions. The number of active clusters indicator in combination with k-means clustering allowed us to identify small events which can be associated with tide influence and water levels. Also, correctly, no major event were identified here.

**LekDijk** Experiments with the LekDijk data, again, showed that the framework can successfully detect minor events. The events we detected, using the number of active clusters – indicator value change

combination (choices made in Steps 2 and 3 in the framework), could be explained by changes in weather, e.g., air temperature, snow, etc. If the boundary number indicator was used, however, all Step 3 methods generated only one category of events, i.e., each point in time was identified as normal, which also makes sense, as the very important changes to be detected are the major ones.

Overall, our experiments show that the three-step change detection framework is a viable option for detecting changes in levee behaviour. The IJkDijk results show that we were able to detect minor and major events, whereas the LiveDijk and LekDijk experiments helped us understanding how to detect minor changes, and how to accommodate for normal behaviour, as well as for unlabelled data.

## 6 Conclusions

In this paper, we have described a framework that can be used to detect changes in the behaviour of levees. Our three-step framework has been examined using three different datasets about various levees. Our results indicate that the framework is effective in detecting changes in levee behaviour, however, more research is needed in order to establish what the best choices of techniques and methods are in the individual steps.

## 7 Acknowledgements

This research has been conducted by IBM in the context of the *Flood Control 2015* program. We would like to thank the IJkDijk foundation, a cooperation between NOM, STOWA, Deltares and TNO, for their support, and Alert Solutions for providing the LekDijk dataset.

## 8 References

- [1] N. Nirupama and S. Simonovic. Increase of Flood-risk due to urbanization: A Canadian example. *Natural Hazards*. 40: 25-41. 2006.
- [2] A. Hooijer, F. Klijn, G. Bas, M. Pedroli and G. Van Os. Towards sustainable flood risk management in the Rhine and Meuse river basins: synopsis of the findings of IRMA-SPONGE. *River Research and Applications*. 20: 343-357. 2004.
- [3] J.K. Vrijling. Probabilistic design of water defense systems in the Netherlands. *Reliability Engineering and System Safety*. 74 : 337-344. 2001.
- [4] Inspectie Verkeer en Waterstaat. *Derde toets primaire waterkeringen*. 2011.
- [5] R.J. Meijer and A.R. Koelewijn. The Development of an Early Warning System for Dike Failures. *The 1<sup>st</sup> International Conference on Waterside Security*. 2008.
- [6] Bezuijzen, A., Kruse, G.A.M. and Van, M.A., Failure of peat dikes in the Netherlands, In *16<sup>th</sup> Int. Conf. Soil Mech. Geot. Eng.* 1857-1860. 2005.
- [7] H. Naruse, Y. Uchiyama, T. Kurashima and S. Unno. River levee Change Detection using distributed fiber optic strain sensor. *IEICE Transactions on Electronics*. 3: 462-476. 2000.

- [8] Y. Beck, A. Khan, P. Cunat, C. Guidoux, O. Artieres, J. Mars and J. Fry. Thermal monitoring of embankment dams by fiber optics. *Proceedings of the ICOLD European Club Symposium on dam safety*. 2010.
- [9] N. Meratnia, B. van der Zwaag, H. van Dijk, D. Bijwaard, P. Havinga. Sensor Networks in the Low Lands. *Sensors*. 10: 8504-8525. 2010.
- [10] V.V. Krzhizhanovskaya, G.S. Shirshov, N.B. Melnikova, R.G. Belleman, F.I. Rusadi, B.J. Broekhuijsen, B.P. Gouldby, J. Lhomme, B. Balis, M.T. Bubak, A.L. Pyayt, I.I. Mokhov, A.V. Ozhigin, B.Lang and R.J. Meijer. Flood early warning system: design, implementation and computational modules. *Proceedings of the International Conference on Computational Science (ICISS)*. 4: 106-115. 2011.
- [11] A.L. Pyayt, I.I. Mokhov, B. Lang, V.V. Krzhizhanovskaya and R.J. Meijer. Machine Learning Methods for Environmental Monitoring and Flood Protection. *International Conference on Artificial Intelligence and Neural Networks*. 337 118-123. 2011.
- [12] F. Maiorana. Performance improvements of a Kohonen self organizing classification algorithm on sparse data sets. *Proceedings of the 10th WSEAS international conference on Mathematical methods, computational techniques and intelligent systems*. 347-352. 2008.
- [13] Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. *SIAM International Conference on Data Mining*,. 389-400, 2009.
- [14] T. Idé and H. Kashima. Eigenspace-based anomaly detection in computer systems. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 440-449. 2004.
- [15] Y. Kawahara, T. Yairi, and K. Machida. Change-point detection in time-series data based on subspace identification. *ICDM'07*. 559-564. 2007.
- [16] V. Guralnik and J. Srivastava. Event detection from time series data. *Knowledge Discovery and Data Mining*. 33-42. 1999.
- [17] R. Fujimaki, T. Nakata, H. Tsukahara, A. Sato, and K. Yamanishi. Mining abnormal patterns from heterogeneous time-series with irrelevant features for fault event detection. *Statistical Analysis and Data Mining*. 2:1-17. 2009.
- [18] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. *Proceedings of the 13th international conference on Very large data bases*. 30:180-191. 2004.
- [19] K. Yamanishi and J. Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. *Knowledge Discovery and Data Mining*. 676-681. 2002.
- [20] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*. 41:15:1-15:58. July 2009.
- [21] M. Y. Kiang. Extending the kohonen self-organizing map networks for clustering analysis. *Computational Statistics & Data Analysis*. 38(2):161-180. 2001.

# Reinforcement Learning for Energy-Reducing Start-Up Schemes

Kristof Van Moffaert

Yann-Michaël De Hauwere  
Ann Nowé

Peter Vrancx

*Computational Modeling Lab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels*  
*{kvmoffae, ydehauwe, pvrancx, anowe} @vub.ac.be*  
*<http://como.vub.ac.be>*

## Abstract

In this paper, we present a learning technique for determining energy-reducing schedules for general devices and equipment. The proposed learning algorithm is based on Fitted-Q Iteration (FQI) and analyzes the usage and the users of a particular device to decide upon the appropriate profile of start-up and shut-down times. We experimentally evaluated our algorithm on a mixture of real-life and simulated data to discover that close-to-optimal control policies can be learned on a short timespan of a only few iterations. Our results show that the algorithm is capable of proposing intelligent schedules that minimize energy consumption and at the same time maximize user satisfaction.

## 1 Introduction

Automatic control policies received a great deal of attention in recent years. Combining such control policies with a low resource consumption and minimal costs is a goal that researchers from various disciplines are attempting to achieve. Traditional self learning or manually designed control strategies lack predictive capabilities to ensure a certain quality of service in systems that are characterized by diverse usage patterns and user preferences. As a result, such systems do not provide effective solutions for achieving the desired resource efficiency. Moreover, such traditional approaches typically also result in a significant risk of temporary discomfort as part of the learning phase or due to ill-configured systems.

In this paper we describe an approach that aims to automatically configure product systems to user demand patterns and their preferences. This means tailoring the performance of devices to the specific circumstances imposed on them by their everyday users. By taking into account patterns in user behavior and expectations, the system usage optimization is twofold. On the one hand side, the quality of service provided by the system to the end user, and on the other hand the resources needed to keep the system running. Such tailoring can be influenced by time dependent usage patterns as well as personal or group determined performance preferences. Both these aspects are brought together in the term *usage profile*.

Consider for instance a coffee machine. A coffee machine has different operational modes: on (making coffee), idle (temporarily heat water) and off. By default, the machine is idle. Every couple of minutes, the machine will re-heat it's water supply, to always be in a state of readiness when a user wants coffee. After office hours, the machine should be turned off manually, to bring down power consumption even further. Bringing the coffee machine from off to idle again in the morning mode requires a warming up phase, which implies that the machine is not immediately usable. On a typical day, the coffee machine used in an office environment will be turned on in the morning and remain on during the day, being used only sporadically. During long periods of time, the machine will be idling. Consistently turning it off after usage is a hindrance because the machine will need to warm up each time it is switched on again. Finding a correct control policy which optimizes energy consumption, without sacrificing human comfort will be the scope of the experiments described later on.

We propose a batch Reinforcement Learning (RL) approach that outputs a control policy based on historic data of usage and user preferences. This approach avoids the overhead and discomfort typically as-

sociated with a learning phase in reinforcement learning while still having the benefit of being adaptive to changing patterns and preferences. In Section 2, we elaborate on related concepts that allow automatic extracting of user patterns. Furthermore, we present our experimental setting in Section 3 and results in Section 4 which are discussed in the subsequent Section 5. To conclude the paper, we form conclusions on the results obtained in Section 6.

## 2 Background and preliminaries

In this section, we focus on related work of techniques concerning automatic retrieval of user patterns and profiles.

### 2.1 MDPs and Reinforcement Learning

A Markov Decision Process (MDP) can be described as follows. Let  $S = \{s_1, \dots, s_N\}$  be the state space of a finite Markov chain  $\{x_t\}_{t \geq 0}$  and  $A = \{a_1, \dots, a_r\}$  the action set available to the agent. Each combination of starting state  $s_i$ , action choice  $a_i \in A_i$  and next state  $s_j$  has an associated transition probability  $T(s_j | s_i, a_i)$  and immediate reward  $R(s_i, a_i)$ . The goal is to learn a policy  $\pi$ , which maps each state to an action so that the the expected discounted reward  $J^\pi$  is maximized:

$$J^\pi \equiv E \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \right] \quad (1)$$

where  $\gamma \in [0, 1]$  is the discount factor and expectations are taken over stochastic rewards and transitions. This goal can also be expressed using Q-values which explicitly store the expected discounted reward for every state-action pair:

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} T(s' | s, a) \max_{a'} Q(s', a') \quad (2)$$

So in order to find the optimal policy, one can learn this Q-function and then use greedy action selection over these values in every state. Watkins described an algorithm to iteratively approximate  $Q^*$ . In the Q-learning algorithm [13] a large table consisting of state-action pairs is stored. Each entry contains the value for  $\hat{Q}(s, a)$  which is the learner's current hypothesis about the actual value of  $Q(s, a)$ . The  $\hat{Q}$ -values are updated accordingly to following update rule:

$$\hat{Q}(s, a) \leftarrow (1 - \alpha_t) \hat{Q}(s, a) + \alpha_t [r + \gamma \max_{a'} \hat{Q}(s', a')] \quad (3)$$

where  $\alpha_t$  is the learning rate at time step  $t$  and  $r$  is the reward received for performing action  $a$  in state  $s$ .

Provided that all state-action pairs are visited infinitely often and a suitable evolution for the learning rate is chosen, the estimates,  $\hat{Q}$ , will converge to the optimal values  $Q^*$  [12].

#### 2.1.1 Fitted-Q Iteration

Fitted Q-iteration (FQI) is a model-free, batch-mode reinforcement learning algorithm that learns an approximation of the optimal Q-function [2]. The algorithm requires a set of input MDP transition samples  $(s, a, s', r)$ , where  $s$  is the transition start state,  $a$  is the selected action and  $s', r$  are the state and immediate reward resulting from the transition, respectively. Given these samples, fitted Q-iteration trains a number of successive approximations to the optimal Q-function in an off-line fashion. The complete algorithm is listed in Algorithm 1. Each iteration of the algorithm consists of a single application of the standard Q-learning update from Equation 3 for each input sample, followed by the execution of a supervised learning method in order to train the next Q-function approximation. In the literature, the fitted Q-iteration framework is most commonly used with tree-based regression methods or with multi-layer perceptrons, resulting in algorithms known as Tree-based Fitted Q-iteration [6] and Neural Fitted-Q iteration [11]. The FQI algorithm is particularly suited for problems with large input spaces and large amounts of data, but where direct experimentation with the system is difficult or costly.

---

**Algorithm 1** Fitted Q-iteration

---

```

 $\hat{Q}(s, a) \leftarrow 0 \quad \forall s, a$  ▷ Initialize approximations
repeat
   $T, I \leftarrow \emptyset$ 
  for all samples  $i$  do ▷ Build training set
     $I \leftarrow I \cup (s_i, a_i)$  ▷ Input values
     $T \leftarrow T \cup r_i + \max_a \hat{Q}(s'_i, a)$  ▷ Target output value
  end for
   $\hat{Q} \leftarrow \text{Regress}(I, T)$  ▷ Train supervised learning method
until Termination
return  $\hat{Q}$  ▷ Return final Q-values

```

---

### 3 Problem setting

In our experiments, we had to model several aspects of the users of a particular household device and the properties of the device itself, e.g. the startup and idle costs. For the relevance of our results, it is important that these models and distributions are as close as possible to the real world. In the sections below, we elaborate into detail how this data was generated.

#### 3.1 Presence and usage probabilities

In the experiments we conducted below, we simulated the presence and usage of six individual users of a household machine. Their presence probability and the average usage of each individual is depicted in Figure 1 and 2, respectively. For the moment, the days we simulate are considered general working days with some deviation in the arrival and departure times of the users. Currently, the devices we are controlling consist of beverage machines, such as a coffee maker or a water dispenser.

For each of the six users, days are generated using these graphs as a probability distribution together with noise added from a Normal distribution. Notice that we also added a user that is present a lot of the time, but does not consume any beverages. It is interesting to notice that for both distribution graphs that during lunch break, both probabilities are diminishing as most people then spend time elsewhere, whereas before and after this timeslot most users tend to be present and drink beverages.

Similar to the distribution concerning presences, we use this distribution as input for a Poisson [7] distribution to generate different usages for different people for different days. The Poisson distribution is especially tailored for expressing the probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with a known average rate and independently of the time since the last event. The combination of the two graphs allow us to generate a variety days with simulated presences and usages up to the level of minutes, i.e. for each simulated day 1440 data points are collected.

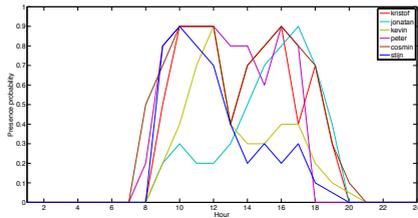


Figure 1: The probability distribution for each of the six individual users on their presence.

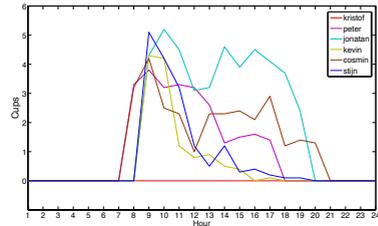


Figure 2: Every user also has a distribution on their usage and drinking behavior.

#### 3.2 A general device model

Another important part of our experimental setting is the model used to represent the device, being controlled. This model should be both general and specific enough to capture all aspects of common household

devices. A Markov Decision Process, as introduced in Section 2.1, is specifically tailored for representing the behavior of a particular household device. In total, two possible actions and three states are presented in an MDP that would cover most, if not all household equipment. The three states or modes of the MDP are 'on', 'off' or 'booting', where the latter represents the time needed before the actual operational mode is reached. The action space  $A$  of our MDP is limited to two distinct, deterministic actions, i.e. the agent can either decide to press a switch or relay that alters the mode of the machine or it can decide do leave the mode of the machine unchanged and do nothing. The former action is a simplification to two separate actions 'turn on' and 'turn off'.

An aspect of the MDP that we did not cover yet is the immediate reward  $R_a(s, s')$  received after transitioning to state  $s'$  from state  $s$  by action  $a$ . These rewards are a combination of two objectives, e.g. an energy consumption penalty or cost and a reward given by the user. The latter is a predefined constant for different situations that can occur. For instance, when the machine is turned off but at the same time a user wanted coffee, then, the current policy does not meet that specific user's profile and the policy is manually overruled. In such a case, the system is provided with a negative feedback signal indicating the user's inconvenience. On the other hand, when the device is turned on at the same time that a user requested a beverage, then the policy actually suits the current user and the system anticipated well on his usage. In those cases, positive reward is provided to the system.

The former reward signal is a measure indicating quality of a certain action  $a$  on the level of power consumption. These rewards are device-dependent and allow the learning algorithm to learn over time whether it is beneficial to have the device in idle mode or if a shutdown is needed. By specifying a certain cost for cold-starting the device, in according to the real-life cost, the algorithm could also learn to power the device on  $x$  minutes before a timeslot with a lot of consumption is expected. In general, the learning algorithm will have to deduct which future timeslots are expected to have a positive difference between the consumption reward signal and the user satisfaction feedback signal. For the moment, these two reward signals are combined into one by a summation.

To conclude, our MDP is graphically represented in Figure 3 and is mathematically formalized as follows:  $M = \langle S, A, P, R \rangle$ , where  $S = \{\text{On}, \text{Off}, \text{Booting}\}$  and  $A = \{\text{Do nothing}, \text{press switch}\}$ . The transitions between the different states are deterministic, resulting in a probability function  $P$  that is shown in Figure 3. The reward function  $R$  is device-specific and we will elaborate this function in the sections below.

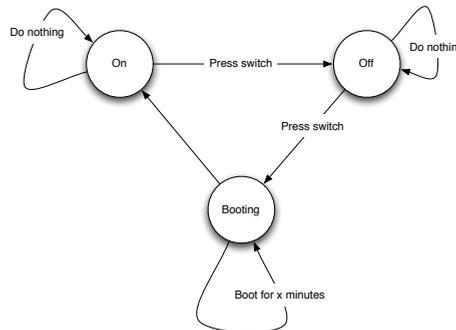


Figure 3: A general model for almost every household device.

## 4 Empirical results

As already mentioned in Section 3, the devices we are focusing on to optimize, are beverage machines such as a coffee maker or a water dispenser. For each of these devices, we use the same MDP  $M$ , depicted in Section 3.2, but we use different reward functions to cover the specific behavior of these machines in real-life. At each of those data points, representing a point in time, the FQI algorithm, described in Section 2.1.1, will decide which action to take from the action space given the current hour, minute and presence set, with 24, 60 and  $2^6$  possible values, respectively. These figures results in a large state space of 92,160 possible combinations. In our setting, the FQI algorithm was first trained with data of one single simulated day and

the control policy was tested for one new day after every training step, whereafter this test sample was also added to the list of training samples to increase the training set's size. Thus, an on-line learning setting was created. In our experiments, we opted for the Tree-Based FQI algorithm with a classification and regression tree or CART [1] and we averaged our results over 10 individual trials.

#### 4.1 Coffee maker experiment

For the first experiment, we mimicked the properties of a real-life coffee maker into our simulation framework. Using appliance monitoring equipment, we have tried to capture the real-life power consumption of the device under different circumstances. After measuring the power consumption of the machine for a few weeks, we came to the following conclusions:

- We noticed that, for our industrial coffee maker, the start-up time was very fast. In just over one minute, the device heated the water up to the boiling temperature and the beverage could be served. The power consumption of actually making coffee is 950 Watts per minute.
- When the machine was running in idle mode, the device is only using around 2 Watts most of the time. However, every ten minutes, the coffee maker re-heated its water automatically. On average, this results in an energy consumption of 50 watts per minute in idle mode.
- The device does not consume any power when turned off.

We normalized these values and incorporated them as rewards and costs into our simulation framework. The results are depicted below. In Figure 4, the average collected reward over 100 simulation days and ten trials is denoted. Starting from an inferior initial policy, the FQI algorithm is able to generalize the large state space and gradually refine its policy. This measure is an indication that indeed energy reducing policies can be learned over time. On the other hand, Figure 5 depicts the number of manual overrides of the proposed policy over each day. We see that in initial runs up to 60 overrides were recorded, where in final runs of the algorithm this is reduced to just under five manual interventions of the user. The combination of both graphs, i.e. the graphs where the collected reward is shown and the one with the number of manual, human interventions, conclude that appropriate schedule can be learned that focus on both energy reduction and user convenience. Figure 6 concludes the experiment on the coffee maker by showing the best policy found by the system. A little before 8 in the morning, before most of the users arrive at the site, the algorithm suggest to turn the machine and leave it on until 14h. Based on the generated days, it was beneficial to have the device turned off for 15 minutes. This could be the case because of the fact that that in the usage data from Figure 2, we have a significant drop in usage at 14h for some heavy drinkers, such as Peter, Kevin and Stijn. Although some people tend to be present until 20h to 22h on some occasions, the algorithm has learned that evening hours are not very beneficial to have the device turned on, because of the low consumption of beverages. We also notice that the proposed policy does not turn the machine on the entire time between 8h and 18h, i.e. there are some minimal moments where the device is turned off for only a few minutes. The reason behind this behavior is two-fold. First, this could be the case because the 100 simulation days did not provide enough information on those particular moments and further learning is needed. Secondly, because the device is up and running in little over one minute, the algorithm can in fact afford to save energy on such a short term as it has observed that the start-up time is minimal.

#### 4.2 Water dispenser experiment

In a second experiment, we tried to perform a similar test where we focused on a different device with different characteristics, i.e. a water dispenser machine. This time, we want to investigate the outcome when different rewards or costs are provided for both the start-up and idle state. More precisely, the start-up costs was divided by four compared to the previous experiment and the idle cost was doubled. Additionally, we extended the start-up time required for the device to be in operational mode from one to ten minutes. We also limited our set of people being monitored to four (i.e. Stijn, Cosmin, Kristof and Kevin).

Similar results as in Section 4.1 were achieved for the collected reward and number of manual overrides of the policy in Figure 7 and 8, respectively. What we do notice is that as a result of the fact that the start-up and the idle cost are approaching each other, the algorithm starts to suggest to shut the device down for particular moments in time. Where in the previous experiment, the device was left on during the lunch break, the algorithm suggested a more conservative schedule with the device turned of for one hour between

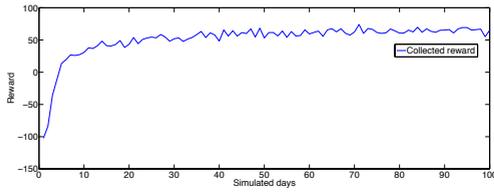


Figure 4: By observing the feedback on the users’ convenience and energy consumption and acting in correspondence of these signals, the policies are being tailored to the users and more reward is being received.

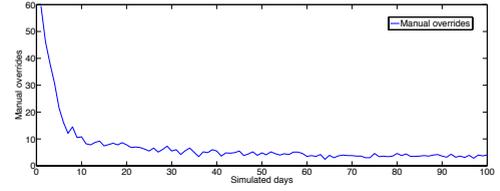


Figure 5: The number of manual interventions and overrides decrease over time as the policy becomes more and more refined to the users and their profiles. In the end, only four manual interventions occur on average.

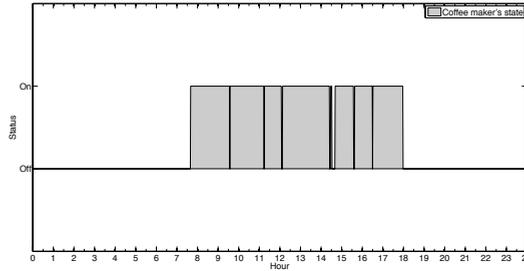


Figure 6: For every point in time, the suggested action of the best policy is graphically represented. The learned policy is specifically suited for a general weekday, where people tend to be present from 9h to 17h with some slight deviations.

12h30 and 13h30. Subsequently, the device is also turned off for a shorter timespan around 11h. Another consequence of these cost functions is the fact that the number of manual overrides (Figure 8) remains quite high because the proposed schedule is now more conservative and meets the user demands less. We also notice from the same figures that the policy is achieved noticeably faster than in the previous experiment, where the number of users being taken into account was larger.

## 5 Discussion and Related Work

In the results, presented in Section 4, we have seen the FQI algorithm in action on a large state space with features corresponding to time-based information and a set of presences. Provided with only a few training points, the algorithm, in combination with its classification and regression tree, managed to obtain acceptable policies after only a few iterations and was able to increase its collected reward while reducing the number of manual interventions needed. Throughout the experiments, we have seen that when providing the model with different reward and cost structures the policies can be heavily influenced. For instance, when an relatively high cost corresponds to leaving the machine running in idle mode, we notice that the algorithm proposed a much more conservative schedule than before and suggests to have the device turned off for longer periods of time.

Previous research has applied the Fitted-Q algorithm mainly in single-objective optimal control problems (e.g. [2, 11, 6]). More recently, [3] also introduced a multi-objective FQI version, which is capable of approximating the Pareto frontier in learning problems with multiple objectives, and applied this algorithm to learn operation policies for water reservoir management. None of these works, however, consider the problem of user interactions and taking into account end-user preferences. To the best of our knowledge this paper presents the first application of FQI in a setting which includes both a cost function and direct user feedback.

Several authors have considered other reinforcement learning algorithms in problem settings related to those presented in this paper. In Dalamagkidis et al. [4], an online temporal difference RL controller is developed to control a building heating system. The controller uses a reinforcement signal which is the weighted combination of 3 objectives: energy consumption, user comfort and air quality. Online RL

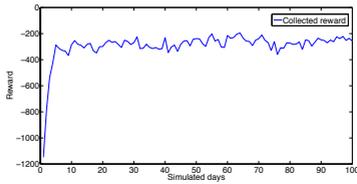


Figure 7: The collected rewards in the water dispenser experiment increased rapidly and the policy converged after 10 iterations on average.

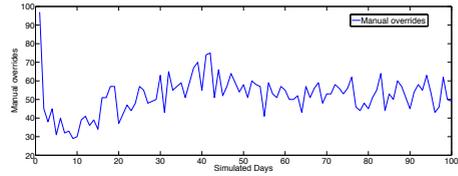


Figure 8: The number of human interventions quickly dropped as more days were being simulated.

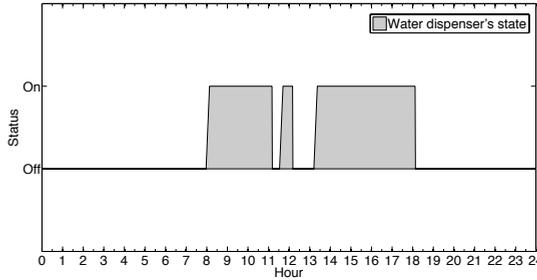


Figure 9: The proposed policy of the algorithm suggests to have the machine on before and after the lunch break. This schedule is more conservative and shuts the device down during noon.

algorithms have also been applied to the problem of energy conservation in wireless sensor networks, often in combination with other objectives such as satisfying certain routing criteria (see e.g. [9, 10, 5]). Finally, Khalili et al. [8] apply Q-learning to learn user preferences in a smart home application setting. Their system is able to adapt to (time-varying) user preferences regarding ambient light and music settings, but does not take into other criteria such as account energy consumption.

## 6 Conclusions

Throughout this paper, we have elaborated the need for algorithms that allow automatic control and optimization of common devices, that at the same time reduce the energy consumption and keep an acceptable level of user convenience. We have elaborated a learning method, based on reinforcement and supervised learning techniques to deduct appropriate start-up and shutdown schedules for common household equipment. A simple model is proposed that allows us to generalize every device and was instantiated to represent two different scheduling cases. For both experiments, the learning agent proposed logical schedules after only a few iterations that focus on both energy reduction and user satisfaction.

In future work, we will focus on taking this framework to the next level and build a real-life setting where the algorithm's policies can be used by a controller device and where they can be gradually refined by manual interventions by real-life users of the system. We will also look into details whether we can combine both reward signals, e.g. the consumption and the user satisfaction feedback signal, in a more intelligent way by, for instance, incorporating techniques from multi-objective optimization.

## Acknowledgements

This research is supported by IWT-SBO project PERPETUAL. (grant nr. 110041).

## References

- [1] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.
- [2] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst. *Reinforcement Learning and Dynamic Programming using Function Approximators*, volume 39 of *Automation and Control Engineering Series*. CRC Press, 2010.
- [3] A. Castelletti, F. Pianosi, and M Restelli. Tree-based fitted q-iteration for multi-objective markov decision problems. In *Proceedings International Joint Conference on Neural Networks (IJCNN 2012)*, 2012.
- [4] K. Dalamagkidis, D. Kolokotsa, K. Kalaitzakis, and G.S. Stavrakakis. Reinforcement learning for energy conservation and comfort in buildings. *Building and Environment*, 42(7):2686 – 2698, 2007.
- [5] M. Devillé, Y-A. Le Borgne, and A. Nowé. Reinforcement learning for energy efficient routing in wireless sensor networks. In P. De Causmaecker, J. Maervoet, T. Messelis, K. Verbeeck, and T. Vermeulen, editors, *Proceedings of the 23rd Benelux Conference on Artificial Intelligence*, pages 89–96, Ghent, Belgium, 2011.
- [6] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.
- [7] F.A. Haight. *Handbook of the Poisson distribution*. Publications in operations research. Wiley, 1967.
- [8] A.H. Khalili, C. Wu, and H. Aghajan. Autonomous learning of users preference of music and light services in smart home applications. In *Proceedings Behavior Monitoring and Interpretation Workshop at German AI Conf.*, 2009.
- [9] Zhenzhen Liu and Itamar Elhanany. A reinforcement learning based mac protocol for wireless sensor networks. *Int. J. Sen. Netw.*, 1(3/4):117–124, January 2006.
- [10] M. Mihaylov, K. Tuyls, and A. Nowé. Decentralized learning in wireless sensor networks. *Lecture Notes in Computer Science*, 4865:60–73, 2010.
- [11] Martin Riedmiller. Neural fitted q iteration – first experiences with a data efficient neural reinforcement learning method. In *In 16th European Conference on Machine Learning*, pages 317–328. Springer, 2005.
- [12] J.N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Journal of Machine Learning*, 16(3):185–202, 1994.
- [13] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, 1989.

# Comparing Context-Aware Routing and Local Intersection Management

Adriaan W. ter Mors

Cees Witteveen

Algorithmics group, Department of Software Technology  
Delft University of Technology

## Abstract

In multi-agent routing, there is a set of mobile agents each with a start location and destination location on a shared infrastructure. An agent wants to reach its destination as quickly as possible, but conflicts with other agents must be avoided. We have previously developed a single-agent route planning algorithm that can find a shortest-time route that does not conflict with any previously made route plans. In this paper, we want to compare this route planning approach with non-planning approaches, in which intersection agents determine which agent may enter an intersection next, and where the agent will subsequently go (given its destination). When making these decisions, the intersection agents use only locally observed traffic information. Our experiments show that context-aware routing produces more efficient results in case no incidents disrupt the execution. However, in the face of unexpected incidents, the performance of the intersection management policies proves very robust, while context-aware routing only produces good results when coupled with effective plan repair mechanisms.

## 1 Introduction

In this paper we will discuss the problem of multi-agent route planning, in which there are multiple mobile agents each with a start and destination location on a roadmap. The roadmap consists of intersections and lanes connecting the intersections, and each agent wants to find a route that will bring it to its destination as quickly as possible.

In previous work [11], we developed a prioritized route planning approach in which agents are first assigned a priority (typically randomly, or based on their arrival time), and subsequently plan a route that is optimal for themselves and does not create any deadlock with any of the higher-priority agents. We named our algorithm *context-aware routing*, as each planning agent is aware of its context, which is the set of reservations from route plans of higher-priority agents. Deadlock prevention is especially relevant in roadmaps with bi-directional roads that can be traversed in only one direction at the same time (e.g., when the roads are not wide enough for two vehicles to travel side by side), for instance in application domains of automated guided vehicles [6] or airport taxi routing [4].

In this paper, however, we will focus on infrastructures in which all roads are directed, such as common in urban traffic control (cf. [1]), and investigate how different routing approaches influence congestion, and therefore the times the agents reach their destinations. We will compare our conflict-free routing approach with a number of local intersection management policies that we will define in section 4. These intersection management policies make routing decisions for the vehicles only on the basis of information that is local to the intersection, namely how many vehicles are waiting to enter the intersection, and how long they have been waiting.

In *urban traffic control*, most intersection management approaches make use of traffic lights, where the focus is on learning efficient behaviour for individual intersections [1]. Coordination is often limited to neighbouring intersections, although the implementation of higher-level agents to support the decision-making is also considered [2]. Another interesting line of work is that into Automated Intersection Management from the group of Peter Stone (see for instance [3]), in which intersections are not light-controlled, but vehicle agents place reservations for conflict-free trajectories in space and time over the intersection. Up until recently, work had focussed on the operation of a single intersection, but recent work by Hausknecht

et al. [5] studies traffic phenomena when multiple intersections are linked together. Vasirani and Osowski [13, 14] propose a market-based approach, in which intersection managers set prices according to current and future demand, and driver agents adapt their routes based on time and cost considerations. Although inspired by Dresner and Stone’s Automated Intersection Management, Vasirani’s research is moving from microscopic models, in which vehicle behaviour is affected by the movements of immediate neighbours, to mesoscopic models in which average traffic densities on roads determine traversal speeds.

This paper contributes to the field of route planning and traffic control by comparing the context-aware route planning approach with local intersection management policies, both in terms of efficiency (measured in e.g. makespan and sum of agent plan costs) and in terms of robustness, i.e., how well the methods perform when unexpected incidents may disrupt the (planned) execution. In addition, we define a set of simple local intersection management policies. Although these policies are not as advanced as some urban traffic management approaches from the literature, we hope that the findings of this research will still hold value for more advanced traffic management (and route planning) approaches.

In section 2, we first present our model for context-aware routing, and then in section 3 we describe the context-aware route planning algorithm, as well as two plan repair mechanisms that are required when incidents can occur during plan execution. Section 4 presents our intersection management policies, and in section 5 we discuss our experimental results. Section 6 contains the conclusions and the ideas for future work.

## 2 Model

We assume a set  $\mathcal{A}$  of agents each with their own start and destination locations in the infrastructure, which is modelled as a *resource graph*  $G_R = (R, E_R)$ , where resources in  $R$  are both roads and intersections, and  $E_R$  is the set of connections between resources. A resource  $r$  has a capacity  $c(r)$ , denoting the maximum number of agents that can simultaneously make use of the resource, and a duration  $d(r) > 0$  which represents the minimum time it takes for an agent to traverse the resource. An agent’s plan consists of a sequence of resources, and a corresponding sequence of intervals in which to visit them.

In this paper, we will restrict ourselves to (non-toroidal) *grids*, where two uni-directional lanes connect each pair of adjacent intersections. For these infrastructures, intersection resources have unit capacity and lane resources have capacity 8; minimum traversal times are 2 time units for the intersections and 7 for the lanes. In previous work (e.g. [10]), we have focused on bi-directional lanes, i.e., lanes on which travel in both directions is possible, though *not at the same time*. In such a setting, however, the local intersection management policies we will evaluate in this paper would cause a deadlock almost instantly. In our context-aware routing approach, deadlocks are prevented by ensuring that agents never make plans that exceed the resource capacities.

**Definition 1 (Resource load)** *Given a set  $\Pi$  of agent plans and the set of all time points  $T$ , the resource load  $\lambda$  is a function  $\lambda : R \times T \rightarrow \mathbb{N}$  that returns the number of agents occupying a resource  $r$  at time point  $t \in T$ :  $\lambda(r, t) = |\{\langle r, \tau \rangle \in \pi \mid \pi \in \Pi \wedge t \in \tau\}|$*

An agent may only make use of a resource in time intervals when the resource load is less than the capacity of the resource. In such a *free time window*, an agent can enter a resource without creating a conflict with any of the existing agent plans.

**Definition 2 (Free time window)** *Given a resource-load function  $\lambda$ , a free time window on resource  $r$  is a maximal interval  $w = [t_1, t_2)$  such that: (i)  $\forall t \in w : \lambda(r, t) < c(r)$ , (ii)  $(t_2 - t_1) \geq d(r)$ .*

Hence, in a free time window there should be both sufficient capacity at any moment during that interval (condition (i)), and it should be long enough for an agent to traverse the resource (condition (ii)). From the free time windows, we can construct a *free time window graph*  $G_W = (W, E_W)$ . For two free time windows to be connected in the graph, their corresponding resources must be connected in the resource graph, and the windows must overlap in time.

The free time window graph encodes the relevant information of the plans of the first  $n - 1$  agents (allowing agent  $n$  to plan its route), but it does not contain any information on the possible movements of agents  $n + j$ ,  $j \geq 1$ . To ensure that agent  $n$  will not make a plan that will make it impossible for any subsequent agent to find a plan, we need to make some simplifying assumptions regarding the start and destination locations of each agent: in this paper, we assume that agents arrive and depart from the infrastructure, like airplanes landing on and taking off from an airport.

### 3 Route Planning Algorithm

In classical shortest path planning, if a node  $v$  is on the shortest path from node  $s$  to node  $t$ , then a shortest path to  $v$  can always be expanded to a shortest path to  $t$ . Figure 1 shows that in prioritized multi-agent route planning, it is not the case that a shortest route to an intermediate resource can always be expanded to the destination: we see a blue agent that wants to go to the rightmost resource, and a black agent that has a plan to travel rightwards at least until the middle intersection. At time 1 (indicated by the numbers inside the vehicles), the blue agent might make a reservation for the leftmost intersection (i.e., slotting in just ahead of the black agent without hindering it), and expand this plan to the middle intersection. From the middle intersection, at time 2, it cannot plan to go right, because that road is momentarily full with vehicles. However, the blue agent must vacate the intersection, because the black agent has a reservation to use it. Hence, the earliest plan to the middle intersection can only be expanded in the upwards direction, which is a detour in space, and possibly time depending on how quickly the grey agents will start moving. The idea

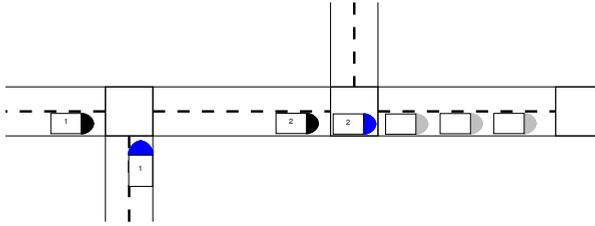


Figure 1: If the blue agent enters the intersection before the black agent, at time 1, then at time 2 it has to drive upwards in order to vacate the intersection for the black agent.

behind our context-aware route planning is that we only need to consider shortest partial plans to the free time windows on a resource: if we have a partial plan that arrives at resource  $r$  at time  $t$  that lies within free time window  $w$ , then all other partial plans to  $r$  that arrive at time  $t'$ , ( $t' \geq t$ )  $\wedge$  ( $t' \in w$ ), can be simulated by waiting in resource  $r$  from time  $t$  to time  $t'$ . Waiting is possible because no conflict will ensue as long as the agent exits  $r$  before the end of  $w$ .

Our route planning algorithm performs a search through the free time window graph that is similar to A\*: In each iteration, we remove a partial plan from an open list of partial plans with a lowest value of  $f = g + h$ , where  $g$  is the actual cost of the partial plan, and  $h$  is a heuristic estimate of reaching the destination resource. In algorithm 1 below, we will write  $\rho(r, t)$  to denote the set of free time windows (directly) reachable from resource  $r$  at earliest exit time  $t$ .

In line 1 of algorithm 1, we check whether there exists a free time window on the start resource  $r_1$  that contains the start time  $t$ . If there is such a free time window  $w$ , then in line 2 we mark this window as **open**, and we record the entry time into  $w$  as the start time  $t$ . In line 5, we select the free time window  $w$  on the **open** list with the lowest value of  $f(w) = g(w) + h(w)$ , where  $g(w)$  is the cost of the partial plan to  $w$ , plus a heuristic estimate  $h(w)$  to reach the destination from  $w$ . If the resource  $r$  associated with  $w$  equals the destination resource  $r_2$ , then we have found the shortest route to  $r_2$ . We return the optimal plan in line 9 by following a series of backpointers.

If  $r$  is not the destination, we expand the plan. First, in line 10, we determine the earliest possible exit time out of  $r$  as the cost of the partial plan:  $g(w) = \text{entryTime}(w) + d(r)$ . Then, in line 11, we iterate over all reachable free time windows that are not **closed**. When expanding free time window  $w$  to free time window  $w'$ , we determine the entry time into  $w'$  as the maximum of the earliest exit time out of resource  $r$ , and the earliest entry time into  $w'$ . We only expand the plan from  $w$  if there has been no previous expansion to free time window  $w'$  with an earlier entry time (initially, we assume that the entry times into free time windows are set to infinity). In line 14, we set the backpointer of the new window  $w'$  to the window  $w$  from which it was expanded. Then, we record the entry time into  $w'$  as  $t_{\text{entry}}$ , and we mark  $w'$  as **open**. Finally, in case no conflict-free plan exists, we return **null** in line 17. The worst-case complexity of algorithm 1 is  $O(|W| \log(|W|) + |E_W|)$ . In case no cyclic plans are allowed, then  $|W| \leq (|\mathcal{A}| + 1)|R|$ , and the complexity of algorithm 1 is  $O(|\mathcal{A}||R| \log(|\mathcal{A}||R|) + |\mathcal{A}||R|^2)$  (proof in [9]). The worst-case complexity of maintaining the free time window graph  $G_W$  is  $O(|\mathcal{A}||R|^2)$ : for each of at most  $R$  reservations of the new plan, one or two new free time windows must be connected to  $O(|W| = |\mathcal{A}||R|)$  existing free time windows.

**Algorithm 1** Plan Route**Require:** start resource  $r_1$ , destination resource  $r_2$ , start time  $t$ ; free time window graph  $G_W = (W, E_W)$ **Ensure:** shortest-time, conflict-free route plan from  $(r_1, t)$  to  $r_2$ .

```

1: if  $\exists w [w \in W \mid t \in \tau_{\text{entry}}(w) \wedge r_1 = \text{resource}(w)]$  then
2:   mark( $w$ , open)
3:   entryTime( $w$ )  $\leftarrow t$ 
4: while open  $\neq \emptyset$  do
5:    $w \leftarrow \text{argmin}_{w' \in \text{open}} f(w')$ 
6:   mark( $w$ , closed)
7:    $r \leftarrow \text{resource}(w)$ 
8:   if  $r = r_2$  then
9:     return followBackPointers( $w$ )
10:   $t_{\text{exit}} \leftarrow g(w) = \text{entryTime}(w) + d(\text{resource}(w))$ 
11:  for all  $w' \in \{\rho(r, t_{\text{exit}}) \setminus \text{closed}\}$  do
12:     $t_{\text{entry}} \leftarrow \max(t_{\text{exit}}, \text{start}(w'))$ 
13:    if  $t_{\text{entry}} < \text{entryTime}(w')$  then
14:      backpointer( $w'$ )  $\leftarrow w$ 
15:      entryTime( $w'$ )  $\leftarrow t_{\text{entry}}$ 
16:      mark( $w'$ , open)
17: return null

```

### 3.1 Plan repair mechanisms

We will now briefly discuss two plan repair mechanisms that can be used to guarantee conflict-free execution for context-aware planners in dynamic environments. The first has been developed by Maza and Castagna [7] and can be considered a baseline approach in the sense that it guarantees conflict-free running without trying to find a repair solution that will result in efficient plan execution. The second is an extension of the first, in which agents can increase their priority over other, delayed agents. Both plan repair mechanisms rely on the fact that, after all agents have made their plans, it is known for each resource (lane or intersection) in which order the agents are scheduled to visit it. The mechanism of Maza and Castagna is simply to adhere to this *resource priority* during plan execution.

In later work, Maza and Castagna developed a repair mechanism that allowed agents to increase their resource priority over delayed agents in such a way that no new deadlocks were introduced [8]. Note that in our current setting, it is not so obvious why such a change in priorities might lead to a deadlock, but for infrastructures with bi-directional resources, attempting a deadlock-free priority change often involves increasing priority over multiple agents for a whole corridor of resources. The second plan repair mechanism we will employ in this paper improves on the algorithm from Maza and Castagna [8] in the sense that it identifies more deadlock-free priority changes, and also leads to a greater reduction in global delay [12].

## 4 Intersection Management

In this section, we will first describe two types of intersection management policies, applied locally at each of the intersections in the infrastructure. The first, most basic type determines which agent is allowed to enter an intersection next, out of the agents ready to enter. The second type of policy then subsequently determines which lane an agent will drive into when it leaves the intersection. In case only an intersection entry policy is employed, the agents follow a randomly chosen shortest path.

**Definition 3 (FCFS)** *Under First-Come First-Served the agent with the earliest entry request time may enter first (ties broken arbitrarily); an agent may request entry once it has reached the intersection.*

One should note that an agent cannot request entry when it is waiting behind another agent; only the first agent in line can request entry. The FCFS policy is simple and fair, but it does not take into account congestion formation on the infrastructure.

**Definition 4 (LQF)** *Under Longest Queue First, the agent that forms the head of the longest queue of vehicles waiting to enter, is allowed to enter.*

Longest Queue First (LQF) aims to reduce congestion in the system by reducing the number of vehicles on the fullest of the roads leading into the intersection. In addition to the roads leading into an intersection, another source of vehicles wanting to enter the intersection is formed by those agents that have their starting point at this intersection. However, this set of vehicles is only counted as a queue of length 1; this means that the LQF policy gives precedence to vehicles already on the infrastructure.

**Definition 5 (WLQF)** Let  $t^*$  be the current time,  $t_i$  the time at which agent  $A_i$  requested entry to the intersection, and  $n_i$  the number of agents on the same road as  $A_i$  at time  $t^*$ . Under Weighted Longest Queue First, the agent that is next to enter is selected according to the formula:

$$\operatorname{argmax}_{i \in \{1, \dots, |A|\}} n_i + f(t^* - t_i) \quad (1)$$

for a given function  $f$ .

In this paper, we have chosen the function  $f$  to divide the argument  $t^* - t_i$  by the minimum travel time of the intersection. Hence, when the function  $f$  returns a value of 5 then it means that a particular agent has been waiting long enough for five agents to traverse the intersection since the time it requested entry.

We will now describe an intersection management policy that directs agents to their next lane resource, which we call the Routing Table Approach (RTA). Under RTA, an intersection will select one of at most three outgoing lanes, thus not including the direction the agent just came from. When an agent enters an intersection, it announces its destination to the intersection agent, which then computes a value for each of the eligible lanes.

**Definition 6 (RTA)** Let  $t^*$  be the time at which agent  $A_k$ , with destination  $z$  is ready to leave the intersection, and let  $L = \{l_1, \dots, l_m\}$ ,  $L \subset R$ , be the eligible outgoing lane resources, and let  $n(l_i)$  denote the number of agents on lane  $l_i$  at time  $t^*$ . Then the Routing Table Approach selects the next lane resource according to the formula:

$$\operatorname{argmin}_{i \in \{1, \dots, |L|\}} g(l_i, z) + \alpha \left( \frac{n(l_i)}{\sum_{j=1}^{|L|} n(l_j)} \right) \quad (2)$$

for some constant  $\alpha$  and function  $g$  that returns the value of the shortest path between its arguments.

In our experiments, we settled on a value of 5.0 for  $\alpha$ ; by comparison, the maximum difference, in our setting, between the road with the shortest path and the road with the longest path was 4. This means that if only one outgoing lane has vehicles on it, then this lane will not be chosen.

## 5 Experimental results

In this section, we describe a set of experiments to comparing the performance of context-aware routing to local intersection management strategies. We mainly look at makespan<sup>1</sup>, but also at sum of agent plan costs, distance travelled, and the number of times an intersection management policy will lead cause a deadlock.

Figure 2 presents the first batch of experiments in which we try increasing numbers of agents on a grid infrastructure of five rows and five columns. Each data point in figure 2(a) is the average of 30 runs, or as many as were completed deadlock-free out of those 30 problem instances. The first conclusion we can draw from figure 2 is that context-aware route planning is invariably faster than intersection management. A second conclusion is that the attempt of the routing table approach to reduce congestion (by selecting a next road with congestion in mind) pays off for two out of three entry policies. RTA combined with Weighted Longest Queue First seems to be the fastest of the local intersection management policies, although there is not much difference with the basic FCFS entry policy.

Figure 2(b) shows, however, that RTA-FCFS and RTA-WLQF are not the best from a completeness point of view; from about 350 to 400 agents, an increasing percentage of experiments results in a deadlock. For FCFS and WLQF, the ability to route agents reduces drastically from about 300 agents; when LQF was employed, however, intersection management had a zero-deadlock rate. The main difference is that WLQF, by taking into account the waiting time of a vehicle wanting to enter the infrastructure, will now and then release a new vehicle into the infrastructure even when long queues have formed at the intersection. The LQF approach, by contrast, will only release a new vehicle when the longest queue of vehicles waiting to enter is at most 1. Hence, using the LQF approach the number of vehicles simultaneously on the infrastructure will be lower, significantly reducing the probability of a deadlock.

<sup>1</sup>All agents have a release time of 0, which means that all agents will either try to obtain a reservation for that time. The makespan is then simply the time at which all agents have reached their destination.

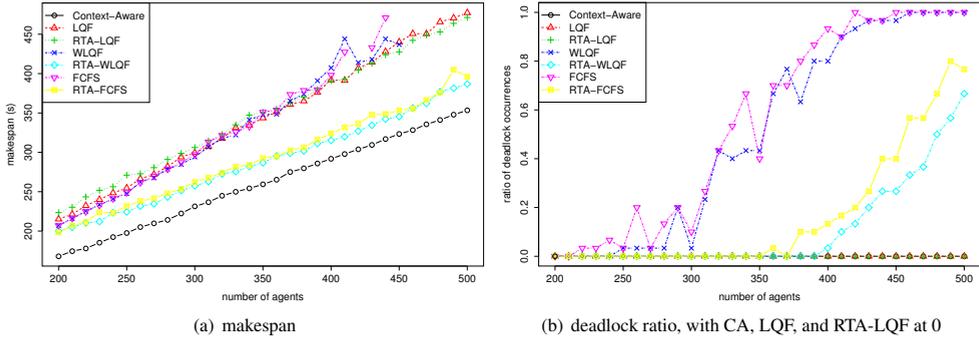


Figure 2: Makespan and percentage of deadlock occurrences on (5, 5) grid infrastructure.

### 5.1 Cost and distance performance measures

We will now briefly look at the results of the experiments from different cost perspectives, in figure 3. In figure 3(a) we see the cost per agent divided by the minimum attainable cost (i.e., the cost of traversing the shortest path when no other agents are around), averaged over all agents. This cost measure is a good indicator of the extent agents suffer from the presence of other vehicles, and we see it increases linearly with the number of agents in the system, regardless of which method is used.

Figure 3(b) shows the distances travelled by each agent (divided by the minimum distance, and averaged over all agents), for each of the methods. For intersection management without RTA, the agents always follow a fixed, and shortest path, so the distance ratio is always 1.0. Agents using context-aware routing may take a slightly longer route if the shortest one is congested, and this results in routes that are on average 5% longer than the shortest path. The most distance is travelled using the routing table approach, as agents are directed away from congested areas. If, however, there is no way around the congested area, then it may happen that agents are kept circling in uncongested areas of the infrastructure until the congestion clears.

Another interesting aspect of figure 3(b) is that for RTA-FCFS and RTA-WLQF, the average distance travelled per agent decreases as the number of agents in the system increases. It seems that, as the system becomes heavily congested, the difference between congestion levels on lanes decreases, and there is no longer any reason to select a longer route.

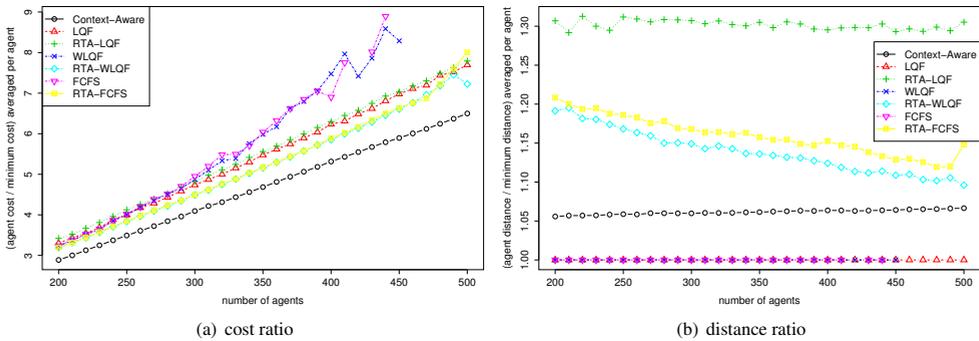


Figure 3: Performance comparison on (5, 5) grid infrastructure, measured in agent cost and distance travelled, divided by a lower bound on cost (and distance).

### 5.2 Unexpected incidents

We will now investigate robustness, i.e., the ability of each of the methods to cope with unexpected delays. We will introduce *vehicle incidents* that render vehicles immobile for a fixed period of time. Incidents are

generated according to a *rate* parameter, which specifies the average number of incidents per vehicle per time unit. Vehicles can only receive incidents when active, i.e., not before they have entered their start location, and not after they have vacated their destination location.

In figure 4, we vary the rate of incidents from 0 to 60 incidents per agent, per hour<sup>2</sup>, and we try two different incident durations: 10 seconds per incident in figure 4(a), and 30 seconds for figure 4(b). All incident-experiments were conducted with 400 agents, about the number of agents for which RTA-WLQF is still able to produce a large percentage of deadlock-free runs. In previous experiments [12, 9, 10], context-

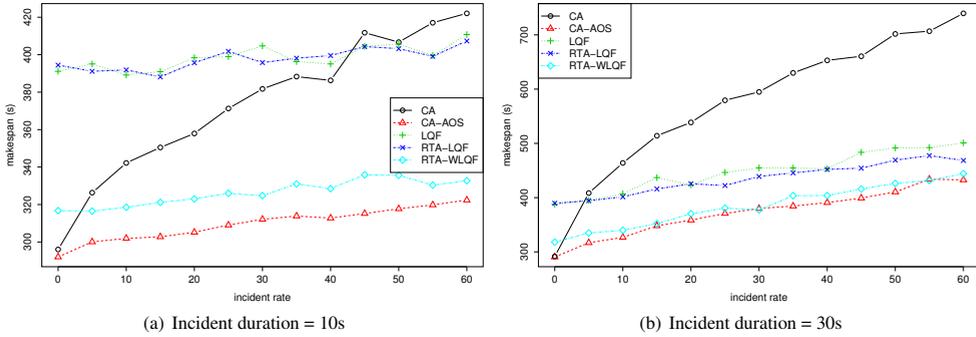


Figure 4: Performance comparison on (5, 5) grid infrastructure with 400 agents and makespan performance measure, with unexpected incidents during the execution.

aware routing approaches were shown to be fairly robust under incidents of this magnitude, but for these types of infrastructures, standard context-aware quickly loses its advantage, especially for longer incidents. An explanation would be that on this type of grid infrastructure, there is a lot of interaction between the agents on a relatively small number of intersections. This means that if one agent is delayed, many other agents have to wait for it. Increasing the priority with the agent order swap mechanism (CA-AOS in figure 4) restores much of the performance of context-aware routing, although for incidents of longer duration it is now almost matched by the best intersection management policies<sup>3</sup>. What is also interesting to note from figure 4 is that the local intersection management policies, and in particular LQF, are very robust in the face of vehicle incidents; although figures 4(a) and 4(b) represent different problem instances (i.e., different pairs of start-and-destination locations), it is interesting to see that the makespan barely increases for longer incidents of 30 seconds. Apparently, when one lane of cars is stuck behind a stricken vehicle, an intersection can use that to simply process more vehicles from the remaining lanes.

## 6 Conclusions and Future Work

In this paper we compared context-aware routing, in which agents sequentially find locally optimal and conflict-free route plans, with local intersection management, in which an intersection agent decides which vehicle is the next to enter, and possibly directs it along the next lane. Our experiments show that, without any incidents disrupting plan execution, context-aware routing produces the most efficient route plans, while only covering on average 5% more distance than always following the shortest path. In case plan interruptions can occur, context-aware routing needs a plan repair mechanism to avoid the situation that many agents are waiting for one delayed agent. With such a repair mechanism in use, there was no longer much difference between the planning approach and the intersection management approach.

One direction for future work is to look into different repair schemes for context-aware routing. The agent order swap mechanism employed in this paper changes the priorities of the agents, but keeps each agent to its originally planned path. Full plan repair, in which an agent computes a completely new route, has been tried in [10] with mixed results. On the one hand, each time an agent successfully makes a new plan it

<sup>2</sup>To put 60 incidents per agent per hour into perspective, note that the total simulation time equals the makespan, which from figure 4 can be seen to vary from around 200 to 700 seconds.

<sup>3</sup>The outcome of the Wilcoxon signed rank test comparing CA-AOS with RTA-WLQF showed that context-aware routing is still significantly better:  $V = 157333$ ,  $p\text{-value} < 2.2 \times 10^{-16}$ ; the 95% confidence interval for the difference in means is [13.99999, 18.00006].

improves its own performance without hindering others (because the new reservations may not conflict with existing ones, adjusted for delays), so full plan repair should be able to improve performance considerably. On the other hand, continual re-planning by all agents has not led to significant global improvement, with agents going back and forth between plans, occasionally covering the same ground multiple times. Hence, a cleverer way of managing the re-planning process is required in order to gain real benefits.

## References

- [1] Ana L. C. Bazzan. A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multi-Agent Systems*, 10:131–164, 2005.
- [2] Ana L.C. Bazzan, Denise de Oliveira, and Bruno C. da Silva. Learning in groups of traffic signals. *Engineering Applications of Artificial Intelligence*, 23(4):560 – 568, 2010.
- [3] Kurt Dresner and Peter Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, pages 591–656, 2008.
- [4] Wolfgang Hatzack and Bernhard Nebel. The operational traffic problem: Computational complexity and solutions. In A. Cesta, editor, *Proceedings of the 6<sup>th</sup> European Conference on Planning (ECP'01)*, pages 49–60, 2001.
- [5] Matthew Hausknecht, Tsz-Chiu Au, and Peter Stone. Autonomous intersection management: Multi-intersection optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, pages 4581–4586, San Francisco, CA, USA, September 25–30 2011. IEEE.
- [6] Chang W. Kim and Jose M.A. Tanchoco. Conflict-free shortest-time bidirectional AGV routing. *International Journal of Production Research*, 29(1):2377–2391, 1991.
- [7] Samia Maza and Pierre Castagna. Conflict-free AGV routing in bi-directional network. In *Proceedings of the 8<sup>th</sup> IEEE International Conference on Emerging Technologies and Factory Automation*, volume 2, pages 761–764, Antibes-Juan les Pins, France, October 2001.
- [8] Samia Maza and Pierre Castagna. A performance-based structural policy for conflict-free routing of bi-directional automated guided vehicles. *Computers in Industry*, 56(7):719–733, 2005.
- [9] A. W. ter Mors. *The world according to MARP*. PhD thesis, Delft University of Technology, March 2010.
- [10] Adriaan W. ter Mors. Conflict-free route planning in dynamic environments. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2166–2171, San Francisco, USA, September 2011.
- [11] Adriaan W. ter Mors, Jonne Zutt, and Cees Witteveen. Context-aware logistic routing and scheduling. In *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling*, pages 328–335, 2007.
- [12] A.W. ter Mors and C. Witteveen. Plan repair in conflict-free routing. In Been-Chian Chien, Tzung-Pei Hong, Shyi-Ming Chen, and Moonis Ali, editors, *Proceedings of the The Twenty Second International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems IEA-AIE 2009*, Lecture Notes in Artificial Intelligence, pages 46–55, Berlin, Heidelberg, June 2009. Springer Verlag LNAI. June 24-27, 2009.
- [13] Matteo Vasirani and Sascha Ossowski. A market-inspired approach to reservation-based urban road traffic management. In K. Decker, J.S. Sichman, C. Sierra, and C. Castelfranchi, editors, *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, volume I, pages 49–56, Richland, SC, May 2009. IFAAMAS. May 10-15, 2009.
- [14] Matteo Vasirani and Sascha Ossowski. A computational market for distributed control of urban road traffic systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):313–321, June 2011.

# Evaluation and Improvement of Laruelle-Widgrén Inverse Banzhaf Approximation<sup>1</sup>

Frits de Nijs

Daan Wilmer

Tomas Klos

*Algorithmics Group, Delft University of Technology  
Delft, The Netherlands*

## Abstract

Voting is a popular way of reaching decisions in multi-agent systems. Weighted voting in particular allows different agents to have varying levels of influence on the decision taken: each agent’s vote carries a *weight*, and a proposal is accepted if the sum of the weights of the agents in favor of the proposal is at least equal to a given *quota*. Unfortunately, there is no clear and unambiguous relation between a player’s weight and the extent of her influence on the outcome of the decision making process. Different measures of ‘power’ have been proposed, such as the Banzhaf and the Shapley-Shubik indices.

Here we consider the ‘inverse’ problem: given a vector of desired power indices for the players, how should we set their weights and the quota such that the players’ power in the resulting game comes as close as possible to the target vector? There has been some work on this problem, both heuristic and exact, but little is known about the approximation quality of the heuristics for this problem. The goal of this paper is to empirically evaluate the heuristic algorithm for the Inverse Banzhaf Index problem by Laruelle and Widgrén. We analyze and evaluate the intuition behind this algorithm. We found that the algorithm can not handle general inputs well, and often fails to improve inputs. It is also shown to diverge after only tens of iterations. Based on our analysis, we present three alternative extensions of the algorithm that do not alter the complexity but can result in up to a factor 6.5 improvement in solution quality.

## 1 Introduction

In systems composed of multiple agents, voting is a popular means of aggregating the preferences of these agents to come to a joint decision. A good example of voting concerns the presidential elections of the United States of America. This follows a two-step process, which nicely illustrates two types of voting. In the first step the citizens vote in each state. Every voter has the same weight and the candidate with the most votes in a state wins that state. The second step illustrates another type of voting: weighted voting. In this step every state votes for the candidate that won in that state. However, it would not be fair if each state has the same vote: the state of California represents over 37 million citizens while little over 560,000 live in Wyoming. Therefore each state has a certain weight, represented by a number of electors. The new president is then chosen by the majority of electors.

With such weighted voting situations, and especially when they are used to elect one of the most powerful men on earth, it is the question how fair the voting is. We can measure this, for example, by using the Banzhaf power index [2] and comparing that to a fair power distribution. Instead of trying to create a fair index—which is much more a philosophical and political question, rather than an algorithmic problem—we try to find a distribution of weights of which the power index matches a target power index.

In Section 2 we give some preliminaries and a more exact definition of the problem we address, along with related work. Section 3 gives a detailed description of the algorithm of Laruelle and Widgrén, including a discussion of some of its weak points. We propose and empirically evaluate alternative methods of initializing the algorithm in Section 4. Then, we propose and evaluate some adaptations to the algorithm itself in Section 5. Section 6 concludes and gives directions for further study.

---

<sup>1</sup>We thank the reviewers for their helpful comments.

## 2 Problem statement

A weighted voting game (WVG) consists of a set  $N$  of  $n$  players  $1, 2, \dots, n$ , each with a voting weight  $w_1, w_2, \dots, w_n$ , along with a *quota*  $q$ . We write a WVG as  $[q; w_1, \dots, w_n]$ . A coalition  $C$  is a subset of players, and every coalition has a value  $v(C) \in \{0, 1\}$  as follows:  $v(C) = 1 \leftrightarrow \sum_{i \in C} w_i \geq q$ . A coalition with value 1 is called *winning*, and a coalition with value 0 is called *losing*.

We are interested in the influence of players on the outcome of decisions in WVGs, i.e., in their so-called ‘voting power.’ To see that a player’s weight is not a good measure of influence, consider the WVG [50; 49, 49, 2]. Here, at least two players are necessary to form a winning coalition, so that in this sense, the third player can be considered to have the same amount of influence as the other two, even though her weight is much lower. A popular—though not the only—method to measure *a priori* power is the Banzhaf power index [2]. It measures the power of a player  $i$  by dividing the number of coalitions of *other* players for which player  $i$  is *critical* (meaning that the coalition is losing, and that player  $i$  can make it winning by joining it), by the total number of coalitions of other players. More formally,

$$\tilde{\beta}_i = \frac{1}{2^{n-1}} \sum_{C \subseteq N \setminus \{i\}} (v(C \cup \{i\}) - v(C)).$$

Often not the regular Banzhaf index is used, but the normalized version [9]. This abstraction is made when it is not interesting *in how many cases* players can actually exert power, but only how the power is *distributed* among players. Because that is what we need, we use the normalized Banzhaf index as well:

$$\beta_i = \frac{\tilde{\beta}_i}{\sum_{j=1}^n \tilde{\beta}_j}.$$

Just computing the power of a weighted voting game is an NP-hard problem [10]. (See [3] for a thorough survey of problems related to power indices and algorithms for solving them.) However, our goal is not to compute power indices, but to solve the ‘inverse’ problem: When we are given a desired distribution of power, we need to find a quota and a set of weights, such that the power in the resulting WVG is distributed as ‘closely’ as possible to the desired distribution (according to some distance measure).

There has been some recent work on the inverse power index problem. De Keijzer, Klos and Zhang [4] propose a method to enumerate all weighted voting games for a given number of agents, that makes use of a partial order that they prove exists on the set of WVGs. By calculating the power distribution for each enumerated game, a game with a power distribution closest to the target will certainly be found. Due to its exponential runtime of  $O(2^{n^2+2n})$ , however, this algorithm is not practical for larger instances—for example for computing the weights for the 27 members of the Council of the EU.

Several other heuristics than the one by Laruelle and Widgrén examined here, have also been proposed. Fatima, Wooldridge and Jennings [7] designed an  $O(n^2)$  approximation algorithm, that traverses the space of WVGs by iteratively shifting parts of the weight from players that have too much power, to players that have too little power, according to a comparison of the game’s power index with the target. Their update rules have a property that makes the algorithm *anytime*: it can be stopped at any iteration and every iteration gives a better or equal result. Unfortunately, it is focused on the Shapley-Shubik power index, which is similar, but not equal to the Banzhaf power index.

Aziz, Paterson and Leech [1] also designed an iterative approximation algorithm. Their algorithm is used to design games that approximate some target Banzhaf index. They use generating functions to calculate the Banzhaf index, which is efficient, but only if the weights are integer. They use interpolation of the current voting power and the desired voting power to determine the next set of weights, multiply them with a certain factor and then round them to integers. The authors don’t analyze the approximation quality of their algorithm.

## 3 Laruelle-Widgrén

In our paper we focus on the iterative algorithm by Laruelle and Widgrén [9]. Starting from a set of initial weights (for which they use the target power index itself), the algorithm iteratively updates the weights of the players by first calculating the Banzhaf index for the given weights, then calculating, for each player,

the ratio of her Banzhaf index and her target power index, and finally dividing each player's weight by its corresponding ratio. That way the weights are adjusted according to the error in the Banzhaf index. The algorithm stops when it is close enough to a given distance threshold, or when a maximum number of iterations has been reached.

A pseudocode version of this algorithm is given in Algorithm 1. It takes as input the valuation function  $v$ , the vector 'target' (also called  $t$  below), the vector  $\omega_0$  containing the initial weights, and the numbers 'maxDistance' (the distance threshold) and 'maxIterations' (the maximum number of iterations). The vectors  $\omega_0$ , weight, banzhafIndex, target and ratio are all vectors of length equal to the number of players.

---

**Algorithm 1** Laruelle-Widgrén ( $v$ , target,  $\omega_0$  (set equal to target in [9]), maxIterations, maxDistance)

---

**Require:** The vector 'target' is a normalized vector of size  $n$ , with  $n > 0$ .

- 1: For each player  $i$ :  $\text{weight}(i) \leftarrow \omega_0(i)$
- 2: iterations  $\leftarrow 0$
- 3: **repeat**
- 4:   banzhafIndex  $\leftarrow \text{calculateBanzhaf}(v, \text{weight})$
- 5:   For each player  $i$ :  $\text{ratio}(i) \leftarrow \frac{\text{banzhafIndex}(i)}{\text{target}(i)}$
- 6:   For each player  $i$ :  $\text{weight}(i) \leftarrow \frac{\text{weight}(i)}{\text{ratio}(i)}$
- 7:   distance  $\leftarrow \text{distance}(\text{banzhafIndex}, \text{target})$
- 8:   iterations  $\leftarrow \text{iterations} + 1$
- 9: **until** (distance < maxDistance)  $\vee$  (iterations > maxIterations)

**Ensure:** weight is a vector of size  $n$

---

The authors do not provide any guarantees for the algorithm. In their paper it is shown to give a good approximation for some cases, but the general case is not analyzed. Here, we do focus on the general case, which is why we refrain from giving the details of the specific WVGs analyzed by Laruelle and Widgrén—those being the WVGs governing different types of decision making in the Council of the EU.

As mentioned above, the original algorithm sets the starting weights  $\omega_0$  (the initial weight vector) to the target vector. Intuitively, this makes sense because the power and the weight distribution are related to some extent. As it turns out, however, this choice can easily lead to selecting starting weights in which at least one of the players has a Banzhaf index of zero. For example, if the highest weight is larger than the quota, then all the players with lower weights have zero power according to the Banzhaf index. Then the ratio for that player will also be zero, resulting in a divide-by-zero error in the first iteration (in line 6, also see [5]). In the next section we present two alternative methods for setting  $\omega_0$  to alleviate this issue, and we evaluate them empirically in section 4.2.

## 4 Alternative $\omega_0$

We consider three methods for setting  $\omega_0$  in our empirical evaluation.

**Target** The initial weight distribution used by Laruelle and Widgrén.

**Centroid** A centroid is a type of center, intuitively defined as the average of all vectors in the body, or alternatively as the center of its mass. As in [4], and without loss of generality, we look only at canonical WVGs, in which the vector of weights is ordered in non-increasing order from player 1 to  $n$ . Thus, all allowed weight vectors appear in what is called the *ordered* simplex. The centroid of a Simplex is computed as the normalized sum of its  $n$  vertices  $(v_0, \dots, v_{n-1})$ . Using this method, the initial weight vector is  $\omega_0 = \frac{1}{n} \sum_{i=0}^{n-1} v_i$ , where the  $v_i$  are the vertices of the (ordered) simplex. The rationale is that this vector is in a region of the Simplex where, according to Kurz [8], many Banzhaf vectors exist. Starting close to many Banzhaf vectors is desirable, because then each iteration is likely to jump to a new Banzhaf vector which in turn results in a slightly different ratio. Changing the ratio often, introduces variance in the direction of update which intuitively leads to better convergence.

**Offset Target** As the target vector  $t$  may lead to a powerless player and the associated problems, we can smooth the initial weight distribution by averaging  $t$  with the uniform weight distribution. In this setting, player  $i$ 's initial weight vector  $\omega_0(i)$  is computed as:  $\omega_0(i) = \frac{t(i)+1/n}{2}$ .

### 4.1 Evaluation Metrics

For our evaluation we use the concept of Manhattan or Taxicab distance  $d_1$ , or  $||\cdot||_1$ . We chose this distance norm because it relates to the results in [8] on the lower bound on the distance between  $t$  and  $\beta_{opt}$ , and because it is cheap to compute.

In order to illustrate our evaluation metrics, Figure 1 presents an example of the distance between  $t$  and a number of related Banzhaf vectors in two dimensions. In the figure, we consider the following Banzhaf vectors:

- $\beta_{opt}$ , an unknown optimal answer. In this figure, it is not equal to  $t$ , which in general it may not be.
- $\beta_{best}$ , a best known (closest) algorithm output found in a database of previously returned Banzhaf vectors.
- $\beta_{alg\ i}$ , the algorithm output for certain parameter settings  $i$ .

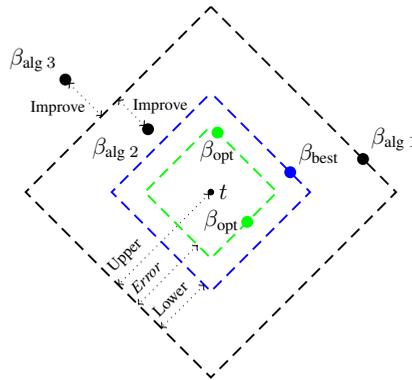


Figure 1: Example of  $d_1$  distance between target vector  $t$  and related Banzhaf vectors in 2D. All points on a diamond are equidistant to  $t$ , according to the  $d_1$  distance metric.

We thus propose two metrics of interest:

1. The *relative improvement* obtained by using a parameter setting 2 compared to 1,  $\frac{d_1(t, \beta_{alg\ 1}) - d_1(t, \beta_{alg\ 2})}{d_1(t, \beta_{alg\ 1})}$ .
2. The *error* in the output produced by a run of the algorithm, defined as  $d_1(t, \beta_{alg\ i}) - d_1(t, \beta_{opt})$ .

The *relative improvement* tells us something about the usefulness of a specific parameter setting. In other words, it tells us how the algorithm should be used to get the best possible results. In figure 1 we can see that the point for  $\beta_{alg\ 2}$  lies inside the black diamond indicating  $d_1(t, \beta_{alg\ 1})$ , so it is an improvement over  $\beta_{alg\ 1}$ . In the computation for improvement we do not compare the points directly, but rather the distance to  $t$ , or the minimum Manhattan distance between  $\beta_{alg\ 2}$  and a point on the diamond for  $\beta_{alg\ 1}$ , shown as ‘Improve’ in figure 1. We consider an improvement to be significant if it exceeds 0.05.

The *error* tells us something about the general usefulness of the algorithm. The error is the minimum Manhattan distance between two points on the green and black diamonds, indicated by the line ‘Error’ in figure 1. Since we do not know  $\beta_{opt}$ , the error must be estimated. An upper bound on the error is  $d_1(t, \beta_{alg\ i})$ . This corresponds to the distance between a point on the black diamond and  $t$ , line ‘Upper’ in figure 1. This upper bound is tight, since it could be that  $t = \beta_{opt}$ , in which case  $d_1(t, \beta_{opt}) = 0$ . A lower bound on the error is  $d_1(t, \beta_{alg\ i}) - d_1(t, \beta_{best})$ . In figure 1 it is the line ‘Lower,’ the minimum Manhattan distance between any two points on the black and blue diamonds. This lower bound is also tight since we may have stored the optimal answer, in which case  $\beta_{best} = \beta_{opt}$ .

The upper bound  $d_1(t, \beta_{alg\ i})$  is a biased estimate of the actual error, since it is in general not the case that  $t = \beta_{opt}$ . This is a consequence of the fact that Banzhaf vectors are composed of rational numbers. Kurz proves in [8] that there exists a lower bound on the largest  $d_1(t, \beta_{opt})$  of  $\frac{1}{9}$ , and conjectures that this bound is actually  $\frac{14}{34}$ . The bias of the lower bound depends on the bias that originates from our method of

obtaining Banzhaf vectors. The quality of the lower bound further depends on the percentage of all Banzhaf vectors we have in our database. We generated Banzhaf vectors by running the algorithm on random samples and storing the vector computed in each iteration of the algorithm (step four) until the database remained constant for 250 consecutive samples. For 8 players this resulted in 1,094,138 Banzhaf vectors, which compared to the 2,730,164 weighted voting games that exists for 8 players [8] means that there exists at least 1 vector for every 2.5 games at this size.

The conjectured lower bound of  $\frac{14}{37}$  on the largest distance between  $t$  and  $\beta_{opt}$  can be used to define the size of a significant error for the upper bound estimate. We say that the upper bound error of an algorithm (parameter setting) is significant if the average value of the error exceeds 10% of  $\frac{14}{37}$ , or  $\frac{7}{185} \approx 0.0378$ . Further, we say a change in the error is significant if the difference exceeds 1% of the maximum value of  $d_1$ . Then a significant error for the lower bound estimate is at least 1% of the maximum value of  $d_1$ .

### 4.2 Experiments with methods for setting $\omega_0$

For our experiments we need a number of target vectors  $t$  to apply the algorithm on. Since a target vector is a vector in the simplex, we produce sample target vectors by drawing a vector uniformly at random from the  $n$ -dimensional ordered Simplex. This can be done by drawing  $n$  samples  $w_i$  from  $U[0, 1]$ , setting  $w_i \leftarrow -\ln w_i$ , renormalizing, and sorting in non-increasing order [6].

In order to evaluate empirically what choice of  $\omega_0$  produces the best results, we performed a number of experiments. For our experiments we drew 10,000 samples from the ordered  $n = 8$ -dimensional Simplex. On each sample we applied the algorithm with all methods for setting the initial weight vector discussed in Section 4, and with  $q = 0.6$  which resulted in the best performance in our initial experiments [5]. Each parameter combination was run for 50 iterations.

Figure 2 presents the results. The x-axis presents the initial distance  $d_1(t, \beta(\omega_0))$  between the target and

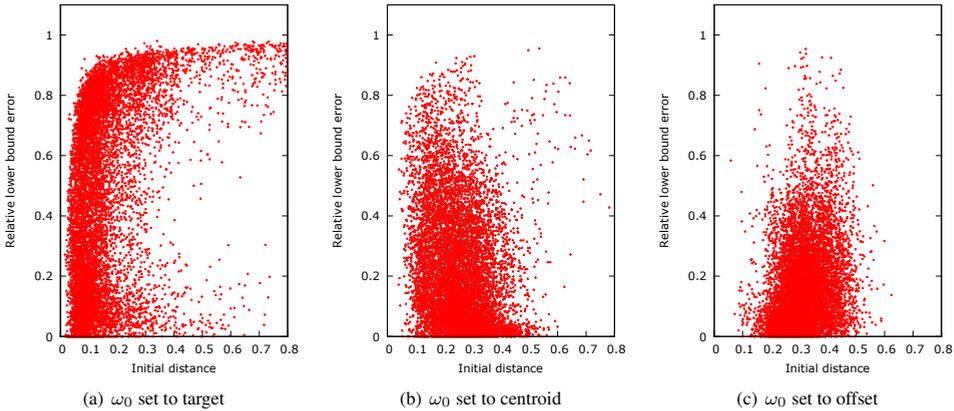


Figure 2: Relative lower bound error compared to the initial distance for the varying  $\omega_0$ .

the power distribution resulting from the initial weight vector  $\omega_0$ . The y-axis shows the lower bound on the error obtained after 50 iterations, relative to the initial distance  $d_1(t, \beta(\omega_0))$ . A lower value means that the algorithm improves more, relative to the initial distance: A value of 1 on the y-axis means that the eventual error found is equal to the initial error. Each datapoint is one of the 10,000 samples.

Table 1 presents the key features (mostly averages) of the data in figure 2. The first column shows the

$\omega_0$ method	Initial	Worst Error	Error Upper	Error Lower
Target	0.1603	0.7988	0.1179	0.0899
Centroid	0.2535	0.6047	0.0803	0.0522
Offset	0.3146	0.4228	0.0833	0.0552

Table 1: Effect of the different choices for  $\omega_0$  on the error.

method for selecting the initial weight vector for the algorithm. The second column shows the initial distance

to  $t$  (averaged over the 10,000 samples), and therefore the average upper bound on the error *before* running the algorithm. The third column presents the worst case distance to  $t$  (across all samples) *after* running the algorithm. Columns four and five present the averages of the upper and lower bounds on the error, where we note the average upper bound on the error is the average distance to  $t$  after applying the algorithm.

These results show how the initial distance is affected by the choice of  $\omega_0$ , with  $\omega_0$  set to the target (Figure 2(a)) having the smallest initial distance and offset (Figure 2(c)) having the largest (also see the second column in Table 1). In this sense, then, using the target as the initial vector was indeed a smart choice. However, when  $\omega_0$  is set to the target, relatively many samples are hard to improve—those with high relative errors. We expect that this is caused by samples starting in a segment with at least one very low target power index, leading it to quickly obtain 0 power, which therefore cannot be improved beyond this initial target, as we explained in section 3. Our intuitions are confirmed by the fact that we don't find this effect when  $\omega_0$  is set by both alternative methods, which were explicitly designed to overcome this. For  $\omega_0$  set to either target or offset a higher initial distance generally results in a higher relative error, while for  $\omega_0$  set to centroid the opposite appears to happen where the higher initial distances are improved more.

The consequence of these hard-to-improve samples is that  $\omega_0$  set as offset or centroid gives a significant improvement in both the upper and lower bound error compared to  $\omega_0$  set to the target. However, the total magnitude of the error is still significant in all cases. The difference between offset and centroid is not significant, but offset does produce the lowest worst case distance to  $t$ , and it can thus be seen as the most robust. Overall we can conclude that starting close to the target is less important than starting in a position where the algorithm can improve the result.

## 5 Algorithmic improvements

In order to factor out the premature stop due to zeros in the weight vector we now propose and evaluate three possible changes to the algorithm itself (rather than to the initialization, as in the previous section).

**Restarts** When a zero is encountered, restart on a new set of weights. This new set cannot be a set that was already encountered, because otherwise the algorithm will not find any new weights and it will stop at the same point, due to the zeros. The new set of weights that we propose is a transformation, in the form of the Offset Target procedure described above, of the best found set of weights to get weights between that point and the centroid of the simplex. Because it is essentially the same as the original algorithm until it stops on a zero, we expect this algorithm to return results that are at least as good as the original, at least beyond the standard number of iterations of the algorithm.

**Coalition** Avoid zeros in the power index by modifying the valuation function such that players without sufficient voting weight can have power. We do this by imposing a minimum coalition size—a feature also of WVGs governing some decision making in the Council of the EU. If there are coalitions with enough voting weight but not enough members, players that have zero voting weight can make it winning by joining it, and therefore have power after all.

**Scaling** Avoid zeros in the power vector by changing the calculation of the new set of weights. Or rather: change the way the ratio is calculated (see Algorithm, 1 line 5). Adding a value, which we call the scaling factor, to both the dividend and the divisor:  $\text{ratio}(i) = \frac{\text{banzhafIndex}(i)+s}{\text{target}(i)+s}$ . As long as  $s > 0$  this will never be reduced to zero, so the weights for a player will always be strictly positive unless the initial weight for that player is zero.

An added effect of this scaling factor is that the steps of the iterations will be smaller, which leads to slower convergence to the target but may also limit overshooting the target.

### 5.1 Evaluation

Since each suggested improvement has its own parameters, we first look at the effect of these parameters in isolation. To examine what relative improvement can be obtained, we drew 10,000 samples from the ordered  $n = 8$ -dimensional simplex and compared the result with that of the standard algorithm run for 50 iterations, with parameters  $\omega_0$  set to the target, and  $q = 0.5$  since we expect this to be the best value in general [5]. For the restart improvement we increased the number of iterations performed from 10 through 80 (original is 50). For the minimum coalition improvement we varied the minimum coalition size from 1 through 8 (original is 1). And finally, for the scaling improvement we varied the scaling factor from 0

through 7 (original is 0). Other parameters are set equal to the original version. The results can be seen in figure 3, which shows relative improvement on the  $y$ -axis (see the definition of this measure in Section 4).

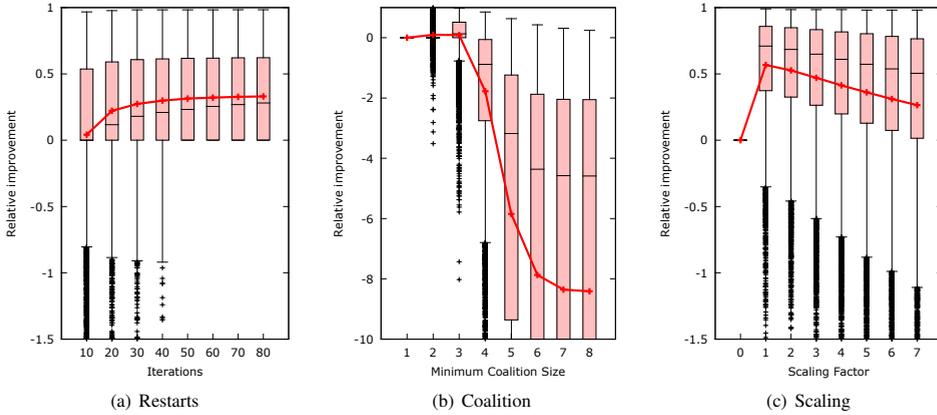


Figure 3: Relative improvement for different parameter settings on the three improvements ( $q = 0.5$  and  $\omega_0$  is set to the target). The figure shows box-plots summarizing 10,000 samples at each parameter setting.

From the figure we can see that restart should be run with the same number of iterations as the base algorithm to obtain universal improvement: If less than 50 iterations are used for the restart algorithm, it may not yet have attained the distance reached by the original algorithm in 50 iterations. For minimum coalition the ideal size appears to be 3, since here the majority of samples is improved. (Note that this is likely related to the dimension in these experiments. For higher numbers of players, we expect the optimum minimum coalition size to increase.) The scaling factor should be positive, but kept small.

To investigate the improvement of the proposed additions in terms of the error, we performed an experiment with tuned parameters for each. Every algorithm was run on 5,000 samples of  $n = 8$ -dimensional Simplex, with parameters set to produce the best results: 50 iterations,  $q = 0.6$  (except when using scaling which performed better with  $q = 0.5$ ), and  $\omega_0$  set to the centroid. Table 2 shows the results. The first

Algorithm	% Improved	% Worse	% Best	Error upper	Error lower
Original	-	-	-	0.0801	0.0520
Restarts	32.9%	0.0%	3.0%	0.0643	0.0363
Coalition (min. 3)	23.6%	18.6%	3.6%	0.0761	0.0481
Scaling ( $s = 0.4$ )	86.2%	12.0%	80.0%	0.0357	0.0078
Sum/Best	-	-	86.5%	0.0344	0.0065

Table 2: Effect of the improvements for ideal parameter settings.

column lists the algorithm under consideration. The second and third columns list the percentage of samples that were improved and made worse, respectively, compared to the base. Column four shows how often an algorithm produced strictly the best result compared to all the others. The fifth and sixth columns present the average distance to the target vector and the best known power vector, respectively. The last row of these columns shows what the results would be like if we could always pick the best algorithm for a sample.

In column four we can see that in 86.5% of the samples precisely one of the improvements produced a game that was closest to  $t$ . (So in 13.5% of all samples, there was not one variant strictly better than the original.) For most samples (80.0%) scaling produced the closest result, however all three improvements have samples they performed best on. The original algorithm never produced a game closest to  $t$  since restarting was given the same number of iterations as the base version, which means it always produced at least the same output. However we can see in column three that both coalition and scaling do produce output that is worse than the base version in more than 10% of the samples.

The table shows that all additions reduce the error on average (in columns five and six, the errors for the three improvements are lower than in the first row of the table), however the magnitude of improvement can only be considered significant for scaling. (Again, significance is established when the error is more than

1% of the maximum  $d_1$  in the simplex (which is 1.75 for  $n = 8$ ) smaller than the error of the original.) Additionally, for scaling, the upper bound on the error is below 10% of the largest possible error, i.e. below  $\frac{7}{185}$  (see section 4.1), and the lower bound is even below 0.5% of the maximum  $d_1$  in the 8-dimensional simplex. Therefore we can say that, by our definition of significance, the error made by scaling is not significantly larger than zero. Compared to the original algorithm with ideal parameters, introducing a scaling factor improves the lower bound error by a factor of more than 6.5.

The other approaches have their own strengths, as restarting does not make the result worse, and minimum coalition actually returns the best result more often than restarting does. In a sense the approaches can be seen to complement each other. If we always take the best result, the improvement compared with scaling is still another 20% in the lower bound which indicates it could be worthwhile to find a new version of the algorithm that combines the effect of the three approaches in some way.

## 6 Conclusion and Future Work

The algorithm by Laruelle and Widgrén works quite well in most cases, but it has some major shortcomings. Our proposals remove the possibility of the algorithm getting stuck in a case where one or more zeroes are in the weight vector, and experiments show that our scaling factor algorithm also improves the average approximation performance. However, in some cases it performs worse than the original algorithm. Our multiple start proposal performs at least as well as the original algorithm, but improves the solution not nearly as much as the scaling factor algorithm. Further work could be done to find an algorithm that improves on these proposals: either by giving a better worst-case performance or by improving the average case approximation, or both.

Our experiments also show that the algorithm is not anytime: an iteration often improves the solution, but it could also deteriorate. Our improvements do not counter that, other than storing the best found solution. This is also something that could be researched in the future. It is also interesting to know the dependence of the improvements of the various proposed modifications on the number of players. So far, we have only evaluated on  $n = 8$  players, but if we want to design voting games for larger numbers of players, we need to know whether they pose additional challenges, and whether we need different designs of our algorithm.

## References

- [1] H. Aziz, M. Paterson, and D. Leech. Efficient algorithm for designing weighted voting games. In *IEEE Multitopic Conference*, pages 1–6, 2007.
- [2] J.F. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Rev.*, 19, 1965.
- [3] B. de Keijzer. A survey on the computation of power indices, and related topics. Technical report, Faculty of EEMCS, Delft University of Technology, 2008.
- [4] B. de Keijzer, T. Klos, and Y. Zhang. Enumeration and exact design of weighted voting games. In *Proceedings AAMAS*, 2010.
- [5] F. de Nijs and D. Wilmer. Evaluation and improvement of Laruelle-Widgrén inverse Banzhaf approximation. <http://arxiv.org/abs/1206.1145>, 2012.
- [6] L. Devroye. *NonUniform Random Variate Generation*. Springer, 1986.
- [7] S. Fatima, M. Wooldridge, and N.R. Jennings. An anytime approximation method for the inverse Shapley value problem. In *Proceedings AAMAS*, 2008.
- [8] S. Kurz. On the inverse power index problem. *Optimization*, 61(8):989–1011, 2012.
- [9] A. Laruelle and M. Widgrén. Is the allocation of voting power among EU states fair? *Public Choice*, 94, 1998.
- [10] K. Prasad and J. S. Kelly. NP-completeness of some problems concerning voting games. *International Journal of Game Theory*, 19, 1990.

# A Tableau-based Reasoning Method for Four-valued Description Logic $\mathcal{ALC}$

Wenzhao Qiao <sup>a</sup>

Nico Roos <sup>a</sup>

<sup>a</sup> *Maastricht University, P.O. Box 616, 6200 MD Maastricht, the Netherlands*

## Abstract

Paraconsistent logic is an attractive approach when dealing with a huge amount of knowledge in an open, dynamic and collaborative environment. It allows users to draw useful conclusions in the presence of inconsistencies. In this sense, introducing paraconsistent reasoning into standard reasoning system not only economizes resources used for fixing contradictions but also maintains proper functioning. In this paper, a four-valued semantics for the description logic  $\mathcal{ALC}$  with a corresponding tableau-based reasoning method is presented. The results demonstrate that this proposed logic collaborating with the tableau-based method has the ability to derive reasonable conclusions from inconsistent knowledge bases.

## 1 Introduction

Ontologies are the structural frameworks for organizing information and are one of the most essential technologies proposed in Semantic Web concept. Usually ontology languages are based on standard description logic (DL) which lacks abilities to handle inconsistent knowledge. However, real knowledge bases are rarely perfect. Inconsistencies occur for several reasons, such as modeling errors when importing ontologies from different sources, errors when merging multi-authored ontologies, and so on. So, it is very important to improve standard semantics to deal with inconsistencies.

There are already approaches presented in the literatures [8, 14, 16] to deal with certain kinds of inconsistencies. These approaches can be classified into two ways. One class of approaches first diagnose and then repair inconsistencies, which could be called “removing inconsistencies”. The other class of approaches employ a non-standard reasoning service to obtain some meaningful results from inconsistent knowledge bases, which could be called “living with inconsistencies”. The first approach is more suited for improving small ontologies. The latter one is better suited for large ontologies in an open environment such as the Web where information is usually distributed.

Current non-standard reasoning services usually follow three different lines,

- Paraconsistent reasoning, which is based on multi-valued logics [6, 11, 12, 14],
- Uncertainty based reasoning, which is based on fuzzy logic or probabilistic logic [8, 19], and
- Inconsistency reasoning, which is based on the selection of consistent subsets [10].

In our previous work [15], we analyzed several existing approaches and their incompatible issue, then proposed a new four-valued semantics for the description logic  $\mathcal{ALC}$ . The reason why we proposed a new semantics for the standard description logic  $\mathcal{ALC}$  is to avoid changing the language of  $\mathcal{ALC}$ . Thus existing knowledge bases and applications can be easily upgraded to this four-valued semantics. This new four-valued semantics for the description logic  $\mathcal{ALC}$  is also applicable to more general description logics such as  $\mathcal{SHOIN}(D_n)$ . The four-valued semantics enables isolation of harmful effects of inconsistencies, and certain kinds of inconsistencies can be eliminated by using two types of preferences. One preference is based on an adaptation of the semantics of the subsumption relation, and the other preference implements specificity.

In this paper, a tableau-based reasoning method for the four-valued description logic is presented. The advantage of a semantic tableaux method is that it is easy to understand. Additionally it is convenient to be

Operator / Concept	Syntax	Semantics
negation	$\neg c$	$\pi^*(\neg c) = O - \pi^*(c)$
conjunction	$c \sqcap d$	$\pi^*(c \sqcap d) = \pi^*(c) \cap \pi^*(d)$
disjunction	$c \sqcup d$	$\pi^*(c \sqcup d) = \pi^*(c) \cup \pi^*(d)$
existential restriction	$\exists r.c$	$\pi^*(\exists r.c) = \{x \in O \mid \exists y \in O, (x, y) \in \pi(r) \text{ and } y \in \pi^*(c)\}$
universal restriction	$\forall r.c$	$\pi^*(\forall r.c) = \{x \in O \mid \forall y \in O, (x, y) \in \pi(r) \text{ implies } y \in \pi^*(c)\}$
everything	$\top$	$\pi^*(\top) = O$
nothing	$\perp$	$\pi^*(\perp) = \emptyset$
subsumption	$c \sqsubseteq d$	$\pi^*(c) \subseteq \pi^*(d)$
equation	$c = d$	$\pi^*(c) = \pi^*(d)$
instance	$a : c$	$\pi(a) \in \pi^*(c)$
	$(a, b) : r$	$(\pi(a), \pi(b)) \in \pi(r)$

Table 1: Syntax and semantics of concept descriptions

implemented in a program. Indeed, semantic tableaux or tableaux style methods exist for various logics [7]. Therefore, the main advantage of developing a tableaux style approach for the four-valued logic is that the new method will be compatible with those existing approaches.

The remainder of this paper is organized as follows. The next section presents the description logic  $\mathcal{ALC}$  and its standard semantics. Section 3 introduces a four-valued semantics for  $\mathcal{ALC}$  and specifies the relation with the standard two-valued semantics. Section 4 introduces a tableau-based reasoning system for the four-valued description logic. Section 5 describes related work, and Section 6 discusses the results and future work.

## 2 Preliminaries

In the description logic  $\mathcal{ALC}$  (Attribute Logic with Complement) [17], we have a set of atomic concepts  $\mathbf{C}$  denoting sets of objects which have certain common properties, a set of individual names  $\mathbf{N}$  denoting objects, and a set of atomic roles  $\mathbf{R}$ . Each atomic role (or role for short) represents a binary relation between two objects.

**Definition 1** Let  $\mathbf{C}$  be a set of atomic concept and let  $\mathbf{R}$  be a set of atomic roles.

The set of concepts  $\mathcal{C}$  is recursively defined as follows:

- $\mathbf{C} \subseteq \mathcal{C}$ ; i.e. atomic concepts are concepts.
- $\top \in \mathcal{C}$  and  $\perp \in \mathcal{C}$ .
- If  $c \in \mathcal{C}$  and  $d \in \mathcal{C}$ , then  $\neg c \in \mathcal{C}$ ,  $c \sqcap d \in \mathcal{C}$  and  $c \sqcup d \in \mathcal{C}$ .
- If  $c \in \mathcal{C}$  and  $r \in \mathbf{R}$ , then  $\exists r.c \in \mathcal{C}$  and  $\forall r.c \in \mathcal{C}$ .
- Nothing else belongs to  $\mathcal{C}$ .

Operators and concepts used in definition 1 are described in table 1, where  $\{c, d\} \subseteq \mathcal{C}$  and  $r \in \mathbf{R}$ . In table 1, relations named “subsumption” and “equation” are concept relations. Assertions specify instances of concepts and relations. A finite set  $\mathcal{T}$  of concept relations is called a TBox. A finite set  $\mathcal{A}$  of assertions is called an ABox. A knowledge base  $\mathcal{K} = (\mathcal{T}, \mathcal{A})$  is a tuple consisting of a TBox  $\mathcal{T}$  and an ABox  $\mathcal{A}$ .

The semantics of description logic is based on a set of interpretations satisfying the knowledge base. An interpretation denoted by  $I = \langle O, \pi \rangle$  consists of a set of objects  $O$  that exist in the world, and an interpretation function  $\pi$ .

**Definition 2** An interpretation  $I = \langle O, \pi \rangle$  is a pair where  $O$  is a non-empty set of objects and  $\pi$  is an interpretation function such that:

- For each atomic concept  $c \in \mathbf{C}$  it holds that  $\pi(c) \subseteq O$ ,
- For each individual  $i \in \mathbf{N}$  it holds that  $\pi(i) \in O$ , and

- For each atomic role  $r \in \mathbf{R}$  it holds that  $\pi(r) \subseteq O \times O$ .

Besides the above defined interpretation, an *extended* interpretation function  $\pi^*$  that specifies the interpretation of concepts in  $\mathcal{C}$  is needed to define the semantics. Table 1 also gives the definition of semantics.

### 3 A Four-valued Semantics for $\mathcal{ALC}$

In this section we introduce a four-valued semantics for the description logic  $\mathcal{ALC}$ . The underlying idea of four-valued semantics is to tolerate inconsistencies by offering additional truth values. The advantage of a four-valued semantics is that the language does not change. We do not have to adapt all the knowledge and information that has been represented using the language. This is especially important when extending the results to the web ontology language OWL.

The four-valued semantics is also described using an interpretation  $I = \langle O, \pi \rangle$  consisting of a set of objects  $O$  and an interpretation function  $\pi$ . In the four-valued semantics, the domain of objects  $O$  is used to construct a bilattice space  $(\{\langle P, N \rangle\}, \leq_t, \leq_k)$ , where  $P$  and  $N$  are subsets of  $O$ . Here the set  $P$  is used to denote the objects that belong to a concept while the set  $N$  is used to denote the objects that belong to the complement of the concept. Note that this enable us to express that information about a concept is incomplete since  $P \cup N = O$  need not hold. Moreover, information about a concept can be contradictory since  $P \cap N = \emptyset$  need not hold. Also note that the bilattice space  $(\{\langle P, N \rangle\}, \leq_t, \leq_k)$  defines two partial orders:  $\leq_t$  and  $\leq_k$ . The partial order  $\leq_t$  expresses differences in the amount of *truth* and the partial order  $\leq_k$  expresses differences in the amount of *information*.

**Definition 3** A four-valued interpretation  $I = \langle O, \pi \rangle$  is a pair where  $O$  is a non-empty set of objects and  $\pi$  is an interpretation function such that:

- For each atomic concept  $c \in \mathbf{C}$  it holds that  $\pi(c) = \langle P, N \rangle$  where  $P, N \subseteq O$ ,
- For each individual  $i \in \mathbf{N}$  it holds that  $\pi(i) \in O$ , and
- For each atomic role  $r \in \mathbf{R}$  it holds that  $\pi(r) \subseteq O \times O$ .

Before we continue formalizing the four valued semantics, we first introduce several useful logical operators on the bilattice. The  $\neg$  operator gives the complement of an element in the bilattice, the  $\wedge$  operator gives the meet of two elements, the  $\vee$  operator the joint, and the superscripts  $\cdot^+$  and  $\cdot^-$  give the projections.

- $\neg$  on direction  $\leq_t$ :  $\neg \langle P, N \rangle = \langle N, P \rangle$ ,
- $\wedge$  and  $\vee$  on direction  $\leq_t$ :  $\langle P_1, N_1 \rangle \wedge \langle P_2, N_2 \rangle = \langle P_1 \cap P_2, N_1 \cup N_2 \rangle$   
 $\langle P_1, N_1 \rangle \vee \langle P_2, N_2 \rangle = \langle P_1 \cup P_2, N_1 \cap N_2 \rangle$
- $\cdot^+$  and  $\cdot^-$ :  $\langle P, N \rangle^+ = P, \langle P, N \rangle^- = N$

Using a four-valued interpretations  $I = \langle O, \pi \rangle$ , we define the interpretation of concepts in  $\mathcal{C}$ .

**Definition 4** The interpretation of a concept  $c \in \mathcal{C}$  is defined by the extended interpretation function  $\pi^*$ .

- $\pi^*(c) = \pi(c)$  iff  $c \in \mathbf{C}$
- $\pi^*(\top) = \langle O, \emptyset \rangle$
- $\pi^*(\perp) = \langle \emptyset, O \rangle$
- $\pi^*(\neg c) = \neg \pi^*(c)$
- $\pi^*(c \sqcap d) = \pi^*(c) \wedge \pi^*(d)$
- $\pi^*(c \sqcup d) = \pi^*(c) \vee \pi^*(d)$
- $\pi^*(\exists r.c) = \langle \{x \in O \mid \exists y \in O, (x, y) \in \pi(r) \text{ and } y \in (\pi(c))^+\}, \{x \in O \mid \forall y \in O, (x, y) \in \pi(r) \text{ implies } y \in (\pi(c))^- \} \rangle$
- $\pi^*(\forall r.c) = \langle \{x \in O \mid \forall y \in O, (x, y) \in \pi(r) \text{ implies } y \in (\pi(c))^+\}, \{x \in O \mid \exists y \in O, (x, y) \in \pi(r) \text{ and } y \in (\pi(c))^- \} \rangle$

In the four-valued semantics, we also use the extended interpretation function  $\pi^*$  to define the truth-values of propositions:  $c \sqsubseteq d$ ,  $a : c$  and  $(a, b) : r$ . The truth-values of the four-valued semantics are defined using sets of the ‘‘classical’’ truth-values:  $t$  and  $f$ . We can create four sets with  $t$  and  $f$  giving us the four truth-values of the four-valued semantics; namely, UNKNOWN:  $\{\}$ , TRUE:  $\{t\}$ , FALSE:  $\{f\}$  and CONFLICT:  $\{t, f\}$ . So, the four-valued TRUE is defined as  $t$  and not  $f$ , and CONFLICT as both  $t$  and  $f$ .

**Definition 5** Let  $\{a, b\} \subseteq \mathbf{N}$  be two individuals, let  $c \in \mathcal{C}$  be a concept and let  $r \in \mathbf{R}$  be a role. Then the interpretation of propositions is defined as:

- $t \in \pi^*(a : c)$  iff  $\pi^*(a) \in \pi^*(c)^+$
- $f \in \pi^*(a : c)$  iff  $\pi^*(a) \in \pi^*(c)^-$
- $t \in \pi^*(c \sqsubseteq d)$  iff  $\pi^*(c)^+ \subseteq \pi^*(d)^+$  and  $\pi^*(d)^- \subseteq \pi^*(c)^-$
- $f \in \pi^*(c \sqsubseteq d)$  iff  $\pi^*(c)^+ \wedge \pi^*(d)^- \neq \emptyset$
- $t \in \pi^*(c = d)$  iff  $\pi^*(c)^+ = \pi^*(d)^+$  and  $\pi^*(d)^- = \pi^*(c)^-$
- $f \in \pi^*(c = d)$  iff  $\pi^*(c)^+ \wedge \pi^*(d)^- \neq \emptyset$  or  $\pi^*(c)^- \wedge \pi^*(d)^+ \neq \emptyset$
- $t \in \pi^*((a, b) : r)$  iff  $(\pi^*(a), \pi^*(b)) \in \pi(r)$
- $f \in \pi^*((a, b) : r)$  iff  $(\pi^*(a), \pi^*(b)) \notin \pi(r)$

Note that we define  $t$  and  $f$  essentially in the same way as we commonly do in standard description logic. The four-valued UNKNOWN, TRUE, FALSE and CONFLICT are the result of using a set.

Using the extended interpretation function of propositions, we can define the entailment relation.

**Definition 6** A four-valued interpretation  $I$  satisfying/entailing a proposition from an ABox or a TBox is defined as:  $I \models^4 \varphi$  iff  $\{t\} \subseteq \pi^*(\varphi)$  (or  $\{t\} \leq_k \pi^*(\varphi)$ )

Note that we use the superscript  $\cdot^4$  to denote the four-valued entailment relation  $\models^4$ . From now on we will denote the classical two-valued entailment relation by  $\models^2$ .

## 4 Semantic Tableaux for the Four-valued Description Logic

This section presents a tableau-based reasoning method for the four-valued description logic. A semantic tableau [18] starts with a set of propositions  $\Gamma$ , which is derived from the knowledge base. Using expansion rules, the original set  $\Gamma$  is expanded to one or more supsets step by step. The objective is to show the negation of the conclusion that we want to verify cannot be satisfied. If any branch meets any *closure condition*, the branch is closed. If all branches close, the proof is finished and the original proposition holds.

We propose a semantic tableaux method for four-valued description logic, which is based on the tableaux method for Belnap’s four-valued logic [7]. Let  $\mathcal{K} = (\mathcal{T}, \mathcal{A})$  be a knowledge base consisting of the TBox  $\mathcal{T}$  and the ABox  $\mathcal{A}$ . Only true propositions are necessarily included in two-valued tableaux. But we will encounter *at least true* and *at least false* propositions besides *true* and *false* ones.

**Definition 7** For a four-valued interpretation  $I = \langle O, \pi \rangle$ ,  $\mathcal{K} \models^4 \alpha$ , labels are defined as:

- $\mathbb{T}\alpha$  iff  $t \in \pi^*(\alpha)$
- $\mathbb{F}\alpha$  iff  $f \in \pi^*(\alpha)$
- $\overline{\mathbb{T}}\alpha$  iff  $t \notin \pi^*(\alpha)$
- $\overline{\mathbb{F}}\alpha$  iff  $f \notin \pi^*(\alpha)$

We wish to verify whether a proposition  $\beta$  is entailed by this knowledge base. Therefore, we start with the set of propositions:  $\Gamma = \mathbb{T}\mathcal{T} \cup \overline{\mathbb{T}}\mathcal{A} \cup \overline{\mathbb{T}}\beta$ . Subsequently,  $\Gamma$  is expanded to one new set  $\Gamma'$  or two new sets  $\Gamma'$  and  $\Gamma''$  in such a way that  $\Gamma$  is satisfiable iff one of these new sets,  $\Gamma'$  or  $\Gamma''$ , is satisfiable. This rewriting process repeats for each new set if the new set is not *closed* and there are still rules that can be applied. A set is *closed* if we can determine that it is unsatisfiable. The tree of sets that we build in this way forms the semantic tableau. If all branches of the semantic tableau are *closed*, the original set  $\Gamma$  is unsatisfiable.

Then we can conclude that  $\beta$  is *at least true*,  $\mathbb{T}\beta$ . Next, a second tableau should be constructed to check the satisfiability of  $\Gamma_0 = \mathbb{T}\mathcal{T} \cup \mathbb{T}\mathcal{A} \cup \overline{\mathbb{F}}\beta$ . If  $\Gamma_0$  is satisfiable,  $\beta$  is not *at least false* which means it is not a conflict proposition, or otherwise.

An important issue is the termination of the construction process of the semantic tableaux. To guarantee termination, infinite long paths should be avoided. A special *blocking* method is used to determine whether a node in the tree of the semantic tableaux contains relevant new information about an individual  $a$ . We say that an individual  $a$  is *blocked* if it is a new individual introduced by one of the rules, it has a role relation  $r$  with an individual  $b$ ,  $(b, a) \in r$  and the information about  $a$  is less or equal to the information about  $b$ . To measure the information about an individual  $a$  we use the function:

$$\Gamma(a) = \{c \in \mathcal{C} \mid a : c \in \Gamma\}$$

In the construction of semantic tableaux, we use the following rewriting rules. Here  $N(\Gamma)$  denotes the set of individual names that occur in the set  $\Gamma$ .

- If  $\mathbb{T}a : c \sqcup d \in \Gamma$ ,  $a$  is *not blocked* and  $\{\mathbb{T}a : c, \mathbb{T}a : d\} \not\subseteq \Gamma$ , then  $\Gamma' = \Gamma \cup \{\mathbb{T}a : c, \mathbb{T}a : d\}$ .
- If  $\mathbb{T}a : c \sqcup d \in \Gamma$ ,  $a$  is *not blocked* and neither  $\mathbb{T}a : c \in \Gamma$  nor  $\mathbb{T}a : d \in \Gamma$  is *not blocked*, then  $\Gamma' = \Gamma \cup \{\mathbb{T}a : c\}$  and  $\Gamma'' = \Gamma \cup \{\mathbb{T}a : d\}$ .
- If  $\mathbb{T}c = d \in \Gamma$  and  $\{\mathbb{T}c \sqsubseteq d, \mathbb{T}d \sqsubseteq c\} \notin \Gamma$ , then  $\Gamma' = \Gamma \cup \{\mathbb{T}c \sqsubseteq d, \mathbb{T}d \sqsubseteq c\}$ .
- If  $\mathbb{T}c \sqsubseteq d \in \Gamma$ ,  $a \in N(\Gamma)$  and neither  $\overline{\mathbb{T}}a : c \in \Gamma$  nor  $\mathbb{T}a : d \in \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{T}}a : c\}$  and  $\Gamma'' = \Gamma \cup \{\mathbb{T}a : d\}$ .
- If  $\mathbb{T}c \sqsubseteq d \in \Gamma$ ,  $a \in N(\Gamma)$  and neither  $\overline{\mathbb{F}}a : d \in \Gamma$  nor  $\mathbb{F}a : c \in \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{F}}a : d\}$  and  $\Gamma'' = \Gamma \cup \{\mathbb{F}a : c\}$ .
- If  $\mathbb{T}a : \exists r.c \in \Gamma$ ,  $a$  is *not blocked* and there is no individual  $x$  such that  $\{\mathbb{T}(a, x) : r, \mathbb{T}x : c\}$ , then  $\Gamma' = \Gamma \cup \{\mathbb{T}(a, b) : r, \mathbb{T}b : c\}$  for some new individual  $b$ .
- If  $\{\mathbb{T}a : \forall r.c, \mathbb{T}(a, b) : r\} \in \Gamma$ ,  $a$  is *not blocked* and  $\mathbb{T}b : c \notin \Gamma$ , then  $\Gamma' = \Gamma \cup \{\mathbb{T}b : c\}$ .
- If  $\mathbb{F}a : c \sqcup d \in \Gamma$ ,  $a$  is *not blocked* and neither  $\mathbb{F}a : c \in \Gamma$  nor  $\mathbb{F}a : d \in \Gamma$ , then  $\Gamma' = \Gamma \cup \{\mathbb{F}a : c\}$  and  $\Gamma'' = \Gamma \cup \{\mathbb{F}a : d\}$ .
- If  $\mathbb{F}a : c \sqcup d \in \Gamma$ ,  $a$  is *not blocked* and  $\{\mathbb{F}a : c, \mathbb{F}a : d\} \not\subseteq \Gamma$ , then  $\Gamma' = \Gamma \cup \{\mathbb{F}a : c, \mathbb{F}a : d\}$ .
- If  $\{\mathbb{F}a : \exists r.c, \mathbb{T}(a, b) : r\} \in \Gamma$ ,  $a$  is *not blocked* and  $\mathbb{F}b : c \notin \Gamma$ , then  $\Gamma' = \Gamma \cup \{\mathbb{F}b : c\}$ .
- If  $\mathbb{F}a : \forall r.c$ ,  $a$  is *not blocked* and there is no individual  $x$  such that  $\{\mathbb{T}(a, x) : r, \mathbb{F}x : c\} \subseteq \Gamma$ , then  $\Gamma' = \Gamma \cup \{\mathbb{T}(a, b) : r, \mathbb{F}b : c\}$  for some new individual  $b$ .
- If  $\overline{\mathbb{T}}a : c \sqcup d \in \Gamma$ ,  $a$  is *not blocked* and neither  $\overline{\mathbb{T}}a : c \in \Gamma$  nor  $\overline{\mathbb{T}}a : d \in \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{T}}a : c\}$  and  $\Gamma'' = \Gamma \cup \{\overline{\mathbb{T}}a : d\}$ .
- If  $\overline{\mathbb{T}}a : c \sqcup d \in \Gamma$ ,  $a$  is *not blocked* and  $\{\overline{\mathbb{T}}a : c, \overline{\mathbb{T}}a : d\} \not\subseteq \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{T}}a : c, \overline{\mathbb{T}}a : d\}$ .
- If  $\overline{\mathbb{T}}c = d \in \Gamma$  and neither  $\overline{\mathbb{T}}c \sqsubseteq d \in \Gamma$  nor  $\overline{\mathbb{T}}d \sqsubseteq c \in \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{T}}c \sqsubseteq d\}$  and  $\Gamma'' = \Gamma \cup \{\overline{\mathbb{T}}d \sqsubseteq c\}$ .
- If  $\overline{\mathbb{T}}c \sqsubseteq d \in \Gamma$  and there is no individual  $x$  such that  $\{\mathbb{T}x : c, \overline{\mathbb{T}}x : d\} \subseteq \Gamma$  or  $\{\mathbb{F}x : d, \overline{\mathbb{F}}x : c\} \subseteq \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{T}}a : c, \overline{\mathbb{T}}a : d\}$  and  $\Gamma'' = \Gamma \cup \{\mathbb{F}a : d, \overline{\mathbb{F}}a : c\}$  for some new individual  $a$ .
- If  $\{\overline{\mathbb{T}}a : \exists r.c, \mathbb{T}(a, b) : r\} \subseteq \Gamma$ ,  $a$  is *not blocked* and  $\overline{\mathbb{T}}b : c \notin \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{T}}b : c\}$ .
- If  $\overline{\mathbb{T}}a : \forall r.c \in \Gamma$ ,  $a$  is *not blocked* and there is no individual  $x$  such that  $\{\mathbb{T}(a, x) : r, \overline{\mathbb{T}}x : c\}$ , then  $\Gamma' = \Gamma \cup \{\mathbb{T}(a, b) : r, \overline{\mathbb{T}}b : c\}$  for some new individual  $b$ .
- If  $\overline{\mathbb{F}}a : c \sqcup d \in \Gamma$ ,  $a$  is *not blocked* and  $\{\overline{\mathbb{F}}a : c, \overline{\mathbb{F}}a : d\} \not\subseteq \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{F}}a : c, \overline{\mathbb{F}}a : d\}$ .
- If  $\overline{\mathbb{F}}a : c \sqcup d \in \Gamma$ ,  $a$  is *not blocked* and neither  $\overline{\mathbb{F}}a : c \in \Gamma$  nor  $\overline{\mathbb{F}}a : d \in \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{F}}a : c\}$  and  $\Gamma'' = \Gamma \cup \{\overline{\mathbb{F}}a : d\}$ .

- If  $\overline{\mathbb{F}}a : \exists r.c \in \Gamma$ ,  $a$  is not blocked and there is no individual  $x$  such that  $\{\mathbb{T}(a, x) : r, \overline{\mathbb{F}}x : c\}$ , then  $\Gamma' = \Gamma \cup \{\mathbb{T}(a, b) : r, \overline{\mathbb{F}}b : c\}$  for some new individual  $b$ .
- If  $\{\overline{\mathbb{F}}a : \forall r.c, \mathbb{T}(a, b) : r\} \in \Gamma$ ,  $a$  is not blocked and  $\overline{\mathbb{F}}b : c \notin \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{F}}b : c\}$ .
- If  $\mathbb{T}a : \neg c \in \Gamma$ ,  $a$  is not blocked and  $\mathbb{F}a : c \notin \Gamma$ , then  $\Gamma' = \Gamma \cup \{\mathbb{F}a : c\}$ .
- If  $\mathbb{F}a : \neg c \in \Gamma$ ,  $a$  is not blocked and  $\mathbb{T}a : c \notin \Gamma$ , then  $\Gamma' = \Gamma \cup \{\mathbb{T}a : c\}$ .
- If  $\overline{\mathbb{T}}a : \neg c \in \Gamma$ ,  $a$  is not blocked and  $\overline{\mathbb{F}}a : c \notin \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{F}}a : c\}$ .
- If  $\overline{\mathbb{F}}a : \neg c \in \Gamma$ ,  $a$  is not blocked and  $\overline{\mathbb{T}}a : c \notin \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{T}}a : c\}$ .

If during the construction of the semantic tableaux, we derive a set  $\Gamma$  containing one of these conditions:

- $\mathbb{T}\alpha, \overline{\mathbb{T}}\alpha$
- $\mathbb{F}\alpha, \overline{\mathbb{F}}\alpha$

We say that this branch is *closed*.

We can prove the following important results for the above described semantic tableaux method.

**Lemma 1** *Let  $\Gamma$  be a finite set of propositions, and  $\Gamma'$  is an immediate extension obtained from  $\Gamma$  by applying tableaux rewriting rules. If  $\Gamma$  is satisfiable, there exists a set  $\Gamma'$  which is satisfiable.*

**Proof** We can prove this lemma by providing the correctness proofs for all the rules described above. We illustrate two representative rules as follows:

- If  $\mathbb{T}a : c \sqcap d \in \Gamma$ ,  $a$  is not blocked and  $\{\mathbb{T}a : c, \mathbb{T}a : d\} \not\subseteq \Gamma$ , then  $\Gamma' = \Gamma \cup \{\mathbb{T}a : c, \mathbb{T}a : d\}$ .  
Correctness:  $\mathbb{T}a : c \sqcap d \in \Gamma$  iff  $t \in \pi^*(a : c \sqcap d)$  iff  $\pi(a) \in \pi^*(c \sqcap d)^+$  iff  $\pi(a) \in \pi^*(c)^+$  and  $\pi(a) \in \pi^*(d)^+$  iff  $\mathbb{T}a : c$  and  $\mathbb{T}a : d$ .
- If  $\overline{\mathbb{T}}a : c \sqcup d \in \Gamma$ ,  $a$  is not blocked and  $\{\overline{\mathbb{T}}a : c, \overline{\mathbb{T}}a : d\} \not\subseteq \Gamma$ , then  $\Gamma' = \Gamma \cup \{\overline{\mathbb{T}}a : c, \overline{\mathbb{T}}a : d\}$ .  
Correctness:  $\overline{\mathbb{T}}a : c \sqcup d \in \Gamma$  iff  $t \notin \pi^*(a : c \sqcup d)$  iff  $\pi(a) \notin \pi^*(c \sqcup d)^+$  iff  $\pi(a) \notin \pi^*(c)^+$  and  $\pi(a) \notin \pi^*(d)^+$  iff  $\overline{\mathbb{T}}a : c$  and  $\overline{\mathbb{T}}a : d$ .

An interpretation satisfies a set  $\Gamma$  iff it satisfies one of the sets  $\Gamma'$  and  $\Gamma''$  that we get by applying a rule of the semantic tableaux method. All the other cases are similar.  $\square$

**Lemma 2** *Let  $\Gamma$  be a finite set of propositions. If  $\Gamma$  is satisfiable, there exists a terminated open tableau starting with  $\Gamma$ . And all the sets in the path from  $\Gamma$  to an open branch are satisfiable.*

**Proof** We start proving by induction to the construction of an open branch whose nodes are satisfiable. The initialization step is given by an immediate extension  $\Gamma'$  or  $\Gamma''$  obtained from  $\Gamma$  which can be satisfiable through lemma 1. Proceeding inductively, it follows there exists a path starting from  $\Gamma$  whose nodes are all satisfiable, and these nodes form an open branch.  $\square$

**Theorem 1 (Sound and Complete)** *Let  $\Gamma$  be a finite set of propositions.  $\Gamma$  is not satisfiable, iff there exists an finite closed tableau starting with  $\Gamma$ .*

**Proof** Suppose  $\Gamma$  is satisfiable, then there is an open tableau starting with  $\Gamma$  through lemma 2 which is contradictory to the premise.

Suppose there is no finite closed tableau starting with  $\Gamma$  that means every terminated tableau has an open branch which cannot be expanded by rewriting rules. This branch represents an interpretation  $I \models^4 \Gamma$ . That is impossible since  $\Gamma$  is not satisfiable.  $\square$

## 5 Related Work

Based on Belnap's four-valued logic system [6], Ofer Arieli and Arnon Avron proposed their propositional language with and four-valued semantics in [1, 2] which provide the foundation for a four-valued description logic. In [11, 12, 13, 14], Yue Ma *et al.* present four-valued description logics with different extensions or reasoning methods. [14] shows a four-valued description logic based on  $SHOIN(D_n)$  whose essential idea is using material implication ( $\rightarrow$ ) to deal with inconsistencies, and two algorithms suitable for implementation are presented in [12]. [11] and [13] respectively extend the four-valued semantics to more expressive description logics,  $SROIQ$  and  $SHIQ$ .

Meanwhile, other kinds of logics, such as fuzzy logic [19] and probabilistic logic [8], are also considered to handle inconsistencies. Diderik Batens presented adaptive logics [4] which have a relation with our work in the sense that these logics also assume the most information is correct. Zhisheng Huang introduced semantic relevant functions to select consistent sub-theories (subsets) to reason with. In general, as we have seen, all the works focus on deploying additional mechanism to achieve reasonable conclusions from inconsistent knowledge bases without eliminating useful conclusions.

Semantic tableaux have been developed for various logics. Anthony Bloesch proposed a signed tableau-style proof system [7] for two paraconsistent logics, Priest's  $\mathcal{LP}$  and Belnap's 4-valued logic. Bernhard Beckert *et al.* provided a tableau-based theorem prover and implemented it in Prolog [5]. Xiaowang Zhang and Zuoquan Lin presented their Quasi-classical logic, and they also chose tableaux for paraconsistent reasoning [20]. Guido Governatori contributed a lot of works for labelled tableaux [9] and non-monotonic reasoning [3].

## 6 Conclusion

In this paper, four-valued semantics for description logic  $\mathcal{ALC}$  is defined, and a corresponding tableau-style reasoning method is introduced. This new four-valued semantics provide a compatible extension to standard description logic  $\mathcal{ALC}$  by avoiding changing the language of  $\mathcal{ALC}$ . Based on this four-valued semantics, developing a tableaux method is the main focus of this paper. All the necessary rewriting rules and the notion of soundness and completeness are presented.

However, some properties of the tableaux still need to be investigated; for example, the complexity issue, and several aspects can be improved. According to our previous works, how to use the semantic tableaux in conflict-minimal interpretation and apply specificity rule also need to be further determined. Although tracking all the premises of conclusions can solve these two problems, it is not an efficient way. Future work will try to address these issues mentioned above, study the relations with well-known paraconsistent and adaptive logics mentioned above. Furthermore, we will address resolving other types of inconsistencies, efficient reasoning services that can handle non-monotonicity (e.g., using argumentation), and the practical evaluation on large (medical) knowledge bases.

## References

- [1] Ofer Arieli and Arnon Avron. Reasoning with logical bilattices. *Journal of Logic, Language and Information*, 5(1):25–63, 1996.
- [2] Ofer Arieli and Arnon Avron. The value of the four values. *Artificial Intelligence*, 102(1):97–141, 1998.
- [3] Alberto Artosi, Guido Governatori, and Antonino Rotolo. Labelled tableaux for nonmonotonic reasoning: Cumulative consequence relations. *Journal of Logic and Computation*, 12(6):1027–1060, 2002.
- [4] Diderik Batens and Joke Meheus. Recent results by the inconsistency-adaptive labourers. In Jean-Yves Bziau, Walter Carnielli, and Dov Gabbay, editors, *Handbook of Paraconsistency*, volume 9 of *Studies in Logic: Logic and Cognitive Systems*, pages 81–99. College Publications, 2007.
- [5] Bernhard Beckert, Reiner Hähnle, Peter Oel, and Martin Sulzmann. The tableau-based theorem prover  $\mathcal{JAP}$ , version 4.0. In M. McRobbie and J. Slaney, editors, *Proceedings of the 13th International Conference on Automated Deduction (CADE)*, New Brunswick, NJ, USA, July 30-August 3, 1996,

- volume 1104 of *Lecture Notes in Computer Science*, pages 303–307. Springer Berlin / Heidelberg, 1996.
- [6] Nuel D. Belnap. A useful four-valued logic. In J. Michael Dunn and George Epstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 8–37. Reidel, Dordrecht, 1977.
- [7] Anthony Bloesch. A tableau style proof system for two paraconsistent logics. *Notre Dame Journal of Formal Logic*, 34(2):295–301, 1993.
- [8] Rosalba Giugno and Thomas Lukasiewicz. P-*SHOQ(D)*: A probabilistic extension of *SHOQ(D)* for probabilistic ontologies in the semantic web. In Sergio Flesca, Sergio Greco, Nicola Leone, and Giovambattista Ianni, editors, *JELIA 2002*, volume 2424 of *Lecture Notes in Computer Science*, pages 86–97. Springer, 2002.
- [9] Guido Governatori. Labelled modal tableaux. In Carlos Areces and Robert Goldblatt, editors, *AiML 2008*, volume 7, pages 87–110. College Publications, 2008.
- [10] Zhisheng Huang and Frank van Harmelen. Using semantic distances for reasoning with inconsistent ontologies. In Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 178–194. Springer Berlin / Heidelberg, 2008.
- [11] Yue Ma and Pascal Hitzler. Paraconsistent reasoning for OWL 2. In Axel Polleres and Terrance Swift, editors, *Web Reasoning and Rule Systems*, volume 5837 of *Lecture Notes in Computer Science*, pages 197–211. Springer Berlin / Heidelberg, 2009.
- [12] Yue Ma, Pascal Hitzler, and Zuoquan Lin. Algorithms for paraconsistent reasoning with OWL. In Enrico Franconi, Michael Kifer, and Wolfgang May, editors, *The Semantic Web: Research and Applications*, volume 4519 of *Lecture Notes in Computer Science*, pages 399–413. Springer Berlin / Heidelberg, 2007.
- [13] Yue Ma, Pascal Hitzler, and Zuoquan Lin. Paraconsistent reasoning for expressive and tractable description logics. In Franz Baader, Carsten Lutz, and Boris Motik, editors, *DL2008*, volume 353 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
- [14] Yue Ma, Zuoquan Lin, and Zhangang Lin. Inferring with inconsistent OWL DL ontology: A multi-valued logic approach. In Torsten Grust, Hagen Höpfner, Arantza Illarramendi, Stefan Jablonski, Marco Mesiti, Sascha Müller, Paula-Lavinia Patranjan, Kai-Uwe Sattler, Myra Spiliopoulou, and Jef Wijsen, editors, *Current Trends in Database Technology - EDBT 2006*, volume 4254 of *Lecture Notes in Computer Science*, pages 535–553. Springer Berlin / Heidelberg, 2006.
- [15] Wenzhao Qiao and Nico Roos. Four-valued description logic for paraconsistent reasoning. In Patrick De Causmaecker, Joris Maervoet, Tommy Messelis, Katja Verbeeck, and Tim Vermeulen, editors, *BANIC 2011*. CODES/KaHo Sint-Lieven, 2011.
- [16] Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In Georg Gottlob and Toby Walsh, editors, *IJCAI-03*, pages 355–362. Morgan Kaufmann, 2003.
- [17] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.
- [18] Raymond Merrill Smullyan. *First-order logic*. Dover Publications, 1995.
- [19] Giorgos Stoilos and Giorgos B. Stamou. Extending fuzzy description logics for the semantic web. In Christine Golbreich, Aditya Kalyanpur, and Bijan Parsia, editors, *OWLED 2007*, volume 258 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
- [20] Xiaowang Zhang and Zuoquan Lin. Paraconsistent reasoning with quasi-classical semantic in  $\mathcal{ALC}$ . In Diego Calvanese and Georg Lausen, editors, *Web Reasoning and Rule Systems*, volume 5341 of *Lecture Notes in Computer Science*, pages 222–229. Springer Berlin / Heidelberg, 2008.

# An Adaptive Stigmergic Coverage Approach for Robot Teams

<sup>1</sup>Bijan Ranjbar-Sahraei

<sup>1</sup>Gerhard Weiss

<sup>2</sup>Ali Nakisaei

<sup>1</sup>{b.ranjbarsahraei,gerhard.weiss}@maastrichtuniversity.nl

*Department of Knowledge Engineering (DKE),*

*Maastricht University, The Netherlands*

<sup>2</sup>ali.n123@gmail.com

*School of Electrical and Computer Engineering,*

*Shiraz University, Shiraz, Iran*

## Abstract

In this paper, an adaptive multi-robot coverage approach called **A-StiCo** (for “Adaptive Stigmergic Coverage”) is described. According to **A-StiCo** multiple robots partition the environment into different regions in an adaptive way and each robot takes responsibility for covering one of these regions. Moreover, the robots communicate indirectly via depositing/detecting pheromones in the environment. Characteristic of **A-StiCo** is that the movement policy for the individual robots is intentionally kept very simple, so that it can be implemented on any unicycle vehicle with minimum computation capability. Crucial for the practical value of any coverage approach is its robustness. Simulation studies are presented which show that **A-StiCo** allows robot teams to fulfill coverage missions in a very efficient and robust way. In particular, the results demonstrate that this approach achieves very robust coverage behavior at the team level under different challenging circumstances (including robot failures and non-convex environments).

## 1 Introduction

In recent years there has been a rapidly growing interest in using teams of mobile robots for covering and patrolling environments of different types and complexities. This interest is mainly motivated by the broad spectrum of potential civilian, industrial and military applications of multi-robot surveillance systems. Examples of such applications are the protection of safety-critical technical infrastructures, the safeguarding of country borders, and the monitoring of high-risk regions and danger zones which cannot be entered by humans in the case of a nuclear incident, a bio-hazard or a military conflict. Triggered by this interest, today automated coverage is a well established topic in multi-robot research which is considered to be of particular practical relevance.

Wagner et al. [19] were the first who suggested to use stigmergic multi-robot coordination for covering/patrolling the environment. They used robots which had the ability to deposit/detect pheromones for modeling an un-mapped environment as a graph, and they proposed to use basic graph search algorithms (such as Depth-First-Search and Breadth-First-Search) for solving robotic coverage problems. Many other researchers used this graph-based modeling scheme in order to design solutions for multi-robot patrolling/covering problems [8, 9, 11, 12, 20]. For example, in [8] Elor and Bruckstein mixed *cycle finding* algorithm with *spreading* algorithm in order to provide a finite-time cycle-based patrolling approach. In contrast to these graph-based techniques, Voronoi-based techniques have recently been introduced for solving robot coverage problems (e.g., see Cortes et al. [3, 4] and Schwager et al. [17, 18]). Based on this idea many researchers have proposed modified covering approaches which are adaptable to changes in the environment and are provably convergent (e.g., [1, 17]). However, the currently available theoretical and algorithmic approaches to multi-robot coverage typically require a group of robots which are capable of direct communication. Additionally, in most cases they also need very complex mathematical computations (e.g., calculating margins and center of mass for an individual Voronoi-region) which also limits their potential

real-world usage. Moreover, many of these methods are based on unrealistic assumptions. Examples of such assumptions are idealized sensors/actuators or sensors with infinite range (e.g. [16]), convexity and/or stationarity of the environment (e.g. [17]), the availability of unlimited communication bandwidth, and fully reliable direct communication links (e.g. [4]).

Recently we proposed a novel stigmergy-based coverage approach called **StiCo** [15] which avoids this type of assumptions. This approach is of a very low computational complexity and is designed for robots with very simple low-range sensors. Moreover, this approach does not rely on direct communication among robots. Instead, the covering robots coordinate on the basis of an indirect communication principle known as stigmergy. In this paper we describe an extended version of **StiCo** called **A-StiCo** (“Adaptive Stigmergic Coverage”). As its name indicates, **A-StiCo** aims at enabling robots to respond *adaptively* to dynamical changes in the environment.

The rest of this article is organized as follows. Section 2 introduces some preliminaries related to the research described here. **A-StiCo** is described in detail in Section 3. Simulation results are shown in Section 4 and Section 5 concludes the article.

## 2 Preliminaries

This section provides some background information on different classes of coverage behavior and on the biological motivation underlying **A-StiCo**. References to relevant related work are given in Section 1.

### 2.1 Coverage Behaviors

In general, a *surveillance* application is characterized by a unique set of requirements. Exploration and Coverage are two fundamental research issues in these applications. Research on exploration and coverage determines how well an area is explored and monitored, respectively. In order to explore the environment, robots decide how to move based on their current information for gaining the most possible new information about the environment. Instead, coverage algorithms determine the spatial relationship of robots in order to optimize some performance functions. In general coverage approaches are classified in three classes [10]: (1) *Blanket coverage*, in which the objective is to achieve a static arrangement of elements that maximizes the detection rate of targets appearing within the coverage area. (2) *Barrier coverage*, in which the objective is to achieve a static arrangement of elements that minimizes the probability of undetected enemy penetration through the barrier. (3) *Sweep coverage*, in which the objective is to move a group of elements across a coverage area in a manner which addresses a specified balance between maximizing the number of detections per time and minimizing the number of missed detections per area. In this paper, the main goal is to achieve a *Blanket coverage* which maximizes the detection rate of the robots. Therefore, robots partition the environment into circular regions, and each robot guards one region by circling around it.

### 2.2 Biological Inspiration

Stigmergy is an indirect form of communication which applies modifications to environment for exchanging information among agents of same species. One of the key features of this kind of communication is its local characteristics where just the immediate neighbors access to the information. Examples of stigmergy can be observed in many kinds of mammals (e.g. rodents, ungulates, carnivores, and prosimians), and many kinds of social insects (e.g. termites, bees, and ants). In particular, many new approaches for problem solving take inspiration from Ant Colonies. Ants use a set of chemicals produced by their living organism (i.e. natural pheromones) that transmit a message to other members of the same species and assists in finding food, locating mate, avoiding danger and help coordinate their social activities.

In computer science, and especially in the field of ant algorithms (e.g., [5]), a number of computational variants of stigmergy have been developed and it has been shown that they allow for very efficient distributed control and optimization in a variety of problem domains (e.g., [6]). In addition to efficiency and distributedness, stigmergy-based coordination has several other properties which are also essential to multi-robot covering algorithms, including robustness, scalability, adaptivity and simplicity.

### 3 Design of the A-StiCo Approach

#### 3.1 Problem Formulation

The basic intention behind the work described here is to design a *motion policy* which enables a group of *robots*, each equipped only with simple *sensors*, to efficiently *cover* a possibly complex *environment*. Moreover, the basic idea pursued is to utilize the principle of pheromone-based coordination and to let each robot deposit *pheromones* on boundaries of its *territory* to inform the others about the already covered areas.

We assume the environment as an *allowable environment* with area  $A$ , where “allowable environment” is defined as a closed and simply connected set which has a finite number of strict concavities [2]. Each robot is a *Dubins* vehicle [7] described by the dynamical system

$$\dot{x} = v \cos \theta, \quad \dot{y} = v \sin \theta, \quad \dot{\theta} = \omega, \quad (1)$$

where  $x, y \in \mathbb{R}$  denote the vehicle position and  $\theta \in \mathbb{S}^1$  denotes its orientation. The control inputs  $v$  and  $\omega$ , describe the forward linear velocity and the angular velocity of the vehicle respectively, while  $v$  is set equal to  $v_0$  (i.e. the nonholonomic vehicle is constrained to move at a constant linear speed) and the control input  $\omega$  takes value in  $[-1/\rho, 1/\rho]$ ;  $1/\rho$  being the maximum curvature.

Each robot is equipped with two ant-antenna like sensors, placed on the front-right and front-left corners. These sensors have the ability to detect presence of pheromones from a predetermined distance called  $R_d$ , where  $R_d$  is considered to be very small. By pheromone, we consider an electrical marker placed at an arbitrary position  $(x_p, y_p)$ . The pheromone is fully evaporated after time  $T_e$ . Inspired by real ants, each robot considers a circular environment of area  $A_T$  as its territory and circles around this area persistently. The area of territory is related to angular and linear velocity of robot as:  $A_T = \pi(v/\omega)^2$ . The motion policy tells a robot what to do at each iteration of time. Therefore, when a robot detects pheromone, it decides based on this policy what to do next. We consider an environment to be covered, as a condition that no two robot territories share a common area of the environment. Therefore, the motion policy should guide the robots in a way that their territory intersections decrease as time passes. When the full coverage is achieved (i.e. no territories have intersection), each robot patrols its territory by moving on the territory border, persistently.

In Fig. 1 a group of robots, moving in an allowable environment are illustrated.

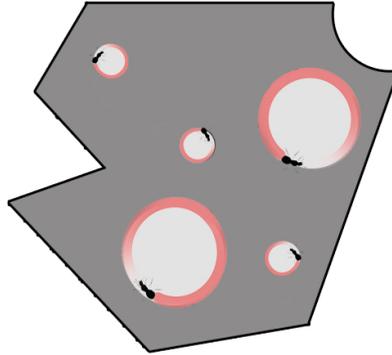


Figure 1: An allowable environment shown in dark gray. Black ants represent the robots, red circle boundaries are the pheromones, and territory of each robot is shown with light gray.

#### 3.2 Basic Motion Policy in StiCo

A-StiCo adopts the basic motion policy of our StiCo approach [15]. In short, StiCo principle is as following: Each robot starts to move with a constant forward linear velocity, and a constant angular velocity, which results in a circular motion on the border of a territory. The forward linear velocity remains constant during the whole mission. However, in different situations the angular velocity might increase or decrease based on the motion policy. Consider one sensor as the interior sensor (the one nearer to the center of territory) and the other one as the exterior sensor. When the interior sensor detects a pheromone, it indicates to the robot

that it is about entering another territory, and therefore the robot changes its circling direction immediately. In this way, the robot establishes its territory in a new region without any intersection with the other territory.

### 3.3 Adaptive Motion Policy in A-StiCo

The **StiCo** coverage approach described in previous subsection works good in various environments. However, when we use a fixed number of robots with fixed territory radius for environments of different sizes, the larger the environment is the less efficient the approach covers the environment. Therefore, to improve the scalability and robustness of this approach, we add an adaptive behavior to the **StiCo** in which robots adjust their angular velocity (and as a consequence their territory area), for efficient coverage of the environment.

Therefore, each robot calculates the time that it has not detected any pheromone. As soon as the time passes a predefined threshold, the robot increases its territory by decreasing the angular velocity (i.e.  $\omega_{new} = \omega_{old} - \Delta\omega$ ). This behavior is illustrated in Fig. 2a. Reversely, when each robot detects pheromones frequently, it decreases its territory area by increasing the angular velocity (i.e.  $\omega_{new} = \omega_{old} + \Delta\omega$ ) as shown in Fig. 2b. Therefore, robots tend to increase their territory area as long as no intersection with other territories happens.

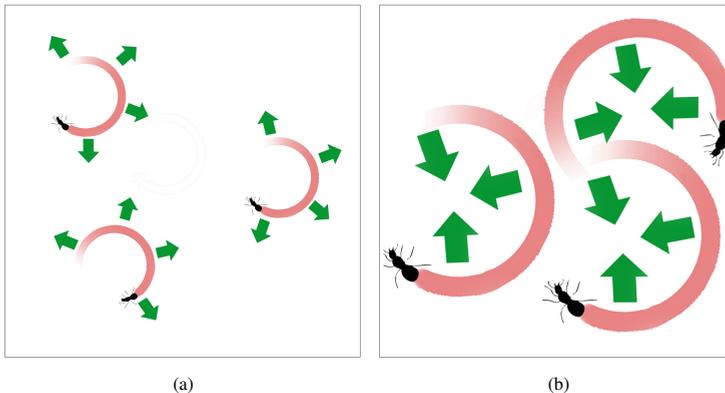


Figure 2: **A-StiCo** adaptation law: (a) when a robot do not detect other robots in nearby, expands its territory area. (b) when a robot detects other robots in nearby, shrink its territory area.

This **A-StiCo** coverage approach is detailed in Algorithm 1 ( $\epsilon$  is the adaptation coefficient used for adjusting the speed of territories expansion).

## 4 Simulation Results

In this section, we demonstrate the evolution of **A-StiCo** on three simulation scenarios. In the first scenario, robots are initialized in the center of an obstacle-free environment and disperse in it homogeneously in order to partition the environment into circular regions. In this simulation, the scalability of **A-StiCo** is demonstrated by using a unique motion policy for robotic swarms of different sizes. In the second scenario, the robust behavior of **A-StiCo** in response to robot failures is illustrated. Finally, in the third simulation, obstacles are used to generate a non-convex coverage problem. The main goal of this scenario is to demonstrate the robustness of **A-StiCo** in complex environments.

All of the simulations are implemented on a robotic swarm of identical members initialized in the center of a  $40m \times 40m$  field. The pheromones are simulated with a high resolution, equal to  $300 \times 300$  and the evaporation time is  $T_e = 1.5s$ . Moreover, we pay careful attention to numerical accuracy and optimization issues in the pheromones update policy.

### 4.1 A-StiCo in a Convex Environment

In this simulation we show that by adding adaptive behavior to the **StiCo** approach, efficient coverage results are achieved. In **A-StiCo**, when a robot does not detect pheromone for a while, it decreases its angular speed

**Algorithm 1** A-StiCo: Adaptive Stigmergic Coverage Approach for an individual robot**Require:** robot can leave and detect pheromone trails

```

1: Initialize: Choose circling direction (CW/CCW)
2: Initialize: Set angular velocity to  $\omega_0$ 
3: loop
4:   while (no pheromone is detected) do
5:     Circle around
6:      $\omega := \omega - \epsilon \cdot \Delta\omega$ 
7:   end while
8:    $\omega := \omega + \Delta\omega$ 
9:   if (interior sensor detects pheromone) then
10:    Reverse the circling direction
11:   else
12:    while (pheromone is detected) do
13:      Rotate
14:    end while
15:   end if
16: end loop

```

( $w_0$ ). Consequently, the territory area is expanded and the robot guards a larger region. Otherwise, when a robot detects pheromone very often (which means that many robots are moving nearby), it increases its angular velocity. Consequently, the territory area becomes smaller and the robot guards a smaller region. Therefore, Robots are able to change their territory area and cover the environment more effectively. Fig. 3 depicts the evolution of **A-StiCo** for two swarms of 10 and 40 robots. In both simulations, robots start from the same initial conditions.

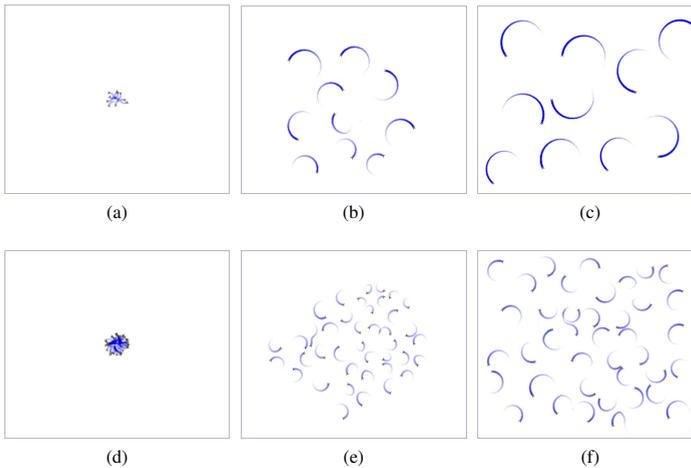


Figure 3: The evolution of **A-StiCo**: (a)-(c) Initial, intermediate, and final snapshots after 250s, for 10 robots. (d)-(f) Initial, intermediate, and final snapshots after 250s, for 40 robots.

## 4.2 Robustness of A-StiCo to Robot Failures

One of the key features of a distributed approach is robustness to individual failures. Therefore, in this scenario we illustrate the efficient behavior of **A-StiCo** in response to a 57% failure of the swarm. Consider a group of 40 robots which are homogeneously positioned in the environment (Fig. 4a). We assume 23 robots are failing to work and eliminate them from this configuration. Due to this large failure in the network, a drastic inhomogeneity is seen which produces vast uncovered regions (Fig. 4b). However, based on adaptive behavior of robots in **A-StiCo** approach, each robot adjusts its territorial area based on the uncovered areas

in its vicinity and it will share the area of its territory with its neighbors. Therefore, soon the network obtain an efficient configuration as shown in Fig. 4c.

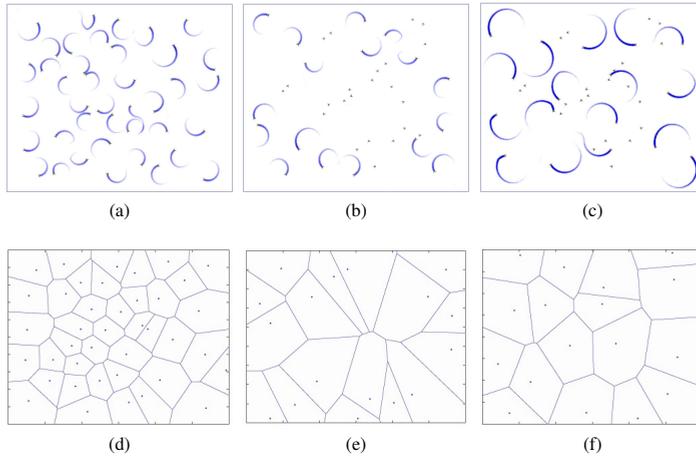


Figure 4: The robustness examination of **A-StiCo** coverage algorithm: (a) Homogeneous coverage after 250s, (b) Failure of 23 robots at  $T=250s$ , (c) Final position after 350s (d-f) Voronoi diagrams of territories centers, each corresponding to upside snapshot.

In order to explore the robust behavior of **A-StiCo** in more detail, we implemented this approach on the same group of robots with different failure percentages (starting form failure of one robot which is 2.5%, up to 39 robot failures which is 97.5%). We defined the recovery time as the time that robots need to adopt their positioning configuration for covering the most possible area of the environment after failures. This recovery time is measured for each simulation, and the results are illustrated in Fig. 5.

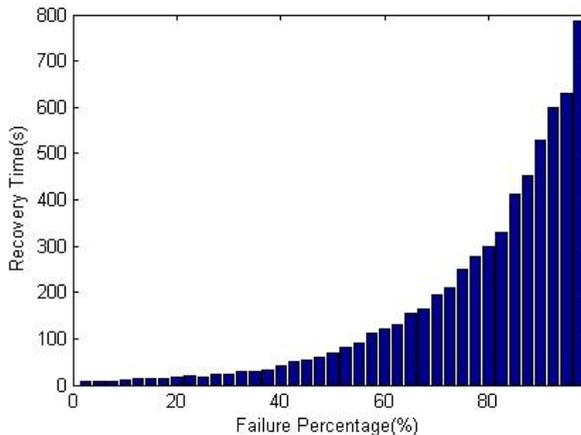


Figure 5: Recovery time for different failure percentages

From the results of Fig. 5, it can be seen that the robotic swarm is very robust to failures up to 60%. However, for larger failures, it takes relatively long time for the robots to adopt their configuration to the new conditions.

### 4.3 Robustness of A-StiCo to Environmental non-Convexities

In this simulation scenario, we consider a non-convex environment as shown in Fig. 6a. This environment can represent a devastated area after an earthquake, or a street map in an emergency condition.

For coverage of this environment, a group of 40 robots are initiated at the center of the environment with different initial angles. **A-StiCo** is executed on this group and snapshots of this simulation are illustrated in Figs 6a-6c. (In this simulation, artificial pheromones are deposited on the borders of obstacles to make them detectable for robots).

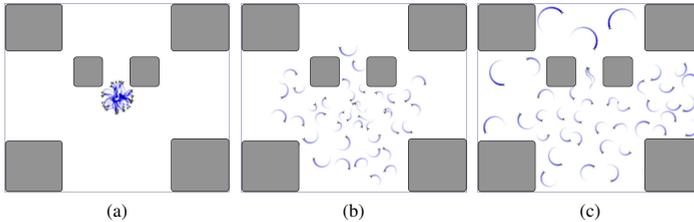


Figure 6: Evolution of **StiCo** in a non-convex environment: (a) Initial snapshot. (b) Intermediate snapshot. (c) Final snapshot.

As shown in the Figs 6a-6c, the **A-StiCo** approach is robust to environmental complexities. Although, robots are not equipped with any path planning system, independent of where the obstacles are placed, robots can easily disperse in the environment homogeneously.

## 5 Conclusion

This article described an extension of the **StiCo** multi-robot coverage approach [15]. **A-StiCo** is a fully distributed motion policy which allows for a very effective and efficient coverage performance. Compared to existing coverage approaches, **A-StiCo** shows several important advantages, including scalability, very low computational complexity and memory requirements, and easy functional extensibility. This makes **A-StiCo** distinct from all other currently available multi-robot coverage approaches. The robust behavior of **A-StiCo** was explored, and simulation results showed that even in the case of robot failures, or environmental complexities, the algorithm performs well.

We think the simulation results justify to invest further research in **StiCo/A-StiCo**. Currently we are working on an implementation of **A-StiCo** on a group of e-puck robots in our SwarmLab (<http://swarmlab.unimaas.nl/>). Based on this experimental test-bed it will be possible to explore the efficiency and robustness of this coverage approach in real-world settings. We also see interesting options for extending **A-StiCo**, and we currently look into the usage of a more advanced pheromone concept. **A-StiCo**, in its current form, does not require that pheromones have the ability of data storage. This makes sense because chemical pheromones (as used by animals) are not appropriate for storing information. However, digital pheromones, (e.g. RFIDs) recently studied in various papers (e.g., [13, 14, 21]) can be used for storing large amount of data in the markers easily.

## References

- [1] A. Breitenmoser, M. Schwager, J. C. Metzger, R. Siegwart, and D. Rus. Voronoi coverage of non-convex environments with a group of networked robots. In *Proc. of the International Conference on Robotics and Automation (ICRA 10)*, pages 4982–4989, May 2010.
- [2] F. Bullo, J. Cortes, and S. Martinez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. 2009. Available at <http://www.coordinationbook.info>.
- [3] J. Cortes, S. Martinez, and F. Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation and Calculus of Variations*, 11:691–719, 2005.

- [4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, 20(2):243 – 255, april 2004.
- [5] M. Dorigo. *Optimization, Learning and Natural Algorithms*. Thesis report, Politecnico di Milano, Italy, 1992.
- [6] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28 –39, nov. 2006.
- [7] L. E. Dubins. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497516, 1957.
- [8] Yotam Elor and Alfred Bruckstein. Autonomous multi-agent cycle based patrolling. In *Swarm Intelligence*, volume 6234 of *Lecture Notes in Computer Science*, pages 119–130. Springer Berlin / Heidelberg, 2010.
- [9] Yotam Elor and Alfred M. Bruckstein. Multi-a(ge)nt graph patrolling and partitioning. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 02, WI-IAT '09*, pages 52–57, Washington, DC, USA, 2009. IEEE Computer Society.
- [10] D. W. Gage. Command control for many-robot systems. In *Proceedings of the Nineteenth Annual AUVS Technical Symposium*, pages 22–24. 1992.
- [11] A. Glad, O. Simonin, O. Buffet, and F. Charpillet. Influence of different execution models on patrolling ant behaviors: from agents to robots. In *Proceedings of the Ninth International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'10)*, 2010.
- [12] Arnaud Glad, Olivier Simonin, Olivier Buffet, and François Charpillet. Theoretical study of ant-based algorithms for multi-agent patrolling. In *Proceeding of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 626–630, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.
- [13] Herianto, Toshiki Sakakibara, and Daisuke Kurabayashi. Artificial pheromone system using rfid for navigation of autonomous robots. *Journal of Bionic Engineering*, 4(4):245 – 253, 2007.
- [14] R. Johansson and A. Saffiotti. Navigating by stigmergy: A realization on an rfid floor for minimalistic robots. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 245 –252, may 2009.
- [15] Bijan Ranjbar-Sahraei, Gerhard Weiss, and Ali Nakisae. Stigmergic coverage algorithm for multi-robot systems (demonstration). In *AAMAS Conference Proceedings*, 2012.
- [16] I. Roman-Ballesteros and C.F. Pfeiffer. A framework for cooperative multi-robot surveillance tasks. In *Electronics, Robotics and Automotive Mechanics Conference, 2006*, volume 2, pages 163 –170, sept. 2006.
- [17] M. Schwager, D. Rus, and J. J. Slotine. Decentralized, adaptive coverage control for networked robots. *International Journal of Robotics Research*, 28(3):357–375, March 2009.
- [18] M. Schwager, D. Rus, and J. J. Slotine. Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment. *International Journal of Robotics Research*, 30(3):371–383, March 2011.
- [19] Israel A. Wagner, Michael Lindenbaum, and Alfred M. Bruckstein. Distributed covering by ant-robots using evaporating traces. *IEEE Transactions on Robotics and Automation*, 15(5):918–933, 1999.
- [20] Vladimir Yanovski, Israel A. Wagner, and Alfred M. Bruckstein. A distributed ant algorithm for efficiently patrolling a network. *Algorithmica*, 37:165–186, 2003.
- [21] V.A. Ziparo, A. Kleiner, L. Marchetti, A. Farinelli, and D. Nardi. Cooperative exploration for USAR robots with indirect communication. In *Proc. of 6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV'07)*, 2007.

# Dynamic Mechanism Design for Efficient Planning under Uncertainty

Joris Scharpff  
*j.c.d.scharpff@tudelft.nl*

Matthijs T.J. Spaan  
*m.t.j.spaan@tudelft.nl*

Mathijs de Weerd  
*m.m.deweerd@tudelft.nl*

*Delft University of Technology  
Mekelweg 4, 2826 CD Delft  
The Netherlands*

## Abstract

We study the planning of maintenance activities on public infrastructural networks – road networks, Internet, power grids, etc. – in contingent environments such that the negative impact on the network user is minimised. Traditional efforts hereto are mainly of a regulatory nature, whereas we propose charging the service-providers (agents) proportional to the harm they cause, thus representing the road user implicitly. Additionally, we seek to exploit the additional opportunities of implicit coordination between agents that arise as a consequence of user cost charging.

In this paper we discuss several existing methods for efficient maintenance planning in contingent environments with interdependent agents and we propose a first attempt at a general dynamic mechanism that is to be refined in future work. By experimental analysis we show the validity of our mechanism.

## 1 Introduction

Recent advances in multi-agent planning bring real-life applications within reach (e.g. [9]). In addition, the field of algorithmic mechanism design has brought computationally feasible methods to make decisions when agents are self-interested [10]. However, in real-life planning problems, such decisions need to be made in a dynamic context and with an uncertain future. For such a complex strategic setting only very few known theoretical results are applicable [1, 2, 3], and we are not aware of any application of these results beyond toy problems (except perhaps [13]). The contribution of this paper is to demonstrate and validate the use of a combination of algorithms for planning with uncertainty [5] and mechanisms for dynamic settings to deal with the problem of coordinating multi-agent planning with interdependent agents in a contingent environment.

This problem is inspired by the real-world setting of maintenance of road infrastructure, such as a national network of highways. Often, several contractors will be responsible for maintaining different segments of the network. Large maintenance activities can impact the throughput of a highway segment: for instance, when renewing the top layer of the road, certain lanes will need to be closed. One of the goals of a national road authority is to ensure that maintenance is planned in a such that the overall throughput of the network is maximized. The crucial element here is that even though contractors are independent firms and maintain different parts of the infrastructure, the maintenance activities (lane closures) of one contractor affect other contractors as traffic flows will change throughout the network.

In the road maintenance problem a national road authority faces a mechanism design problem: how to draw up contracts that incentivise contractors to coordinate their planned maintenance activities so as to minimize the disruption of the overall traffic flow. We model the planning problem of each contractor as a Markov Decision Process (MDP) [5], as it naturally captures the uncertainties that can occur when planning such activities. This paper explores several ways how

these individual problems can be combined and solved, using the theory of dynamic mechanism design.

Below we start by discussing the problem of planning under uncertainty for multiple agents in a bit more detail (Section 2) and outline several solution approaches. We then focus on settings where the agents are self-interested. In Section 3 we discuss the dynamic mechanism design concepts relevant to our work, and present our mechanism using a case study on infrastructural networks in Section 4. Finally, we experimentally verify the validity of our mechanism in Section 5.

## 2 Efficient Planning under Uncertainty

Planning in an ideal world, where all tasks are guaranteed to succeed, is deterministic and hence we are able to develop optimal plans offline. However, in real world applications we are dealing with a contingent environment and tasks might fail due to external factors. This might cause offline optimal plans to result in rather poor results when executed unchanged.

In order to guarantee efficiency throughout the entire execution period, we cannot rely on a single joint plan as it is not robust against contingencies. Instead, we are looking for efficient policies that dictate the action(s) to take given the *current execution state*, whether it was the state we expected beforehand or another state we end up in due to some unforeseen event.

The presence of uncertainty in planning can be captured rather naturally in the Markov Decision Process (MDP) framework[5]. At each point in time the network is in a certain state, for which the agents need to decide what actions to take. The new state of the network depends on the current state and some probabilistic transition function that represents the uncertainty in the executions. Note that this does require knowledge about the possible uncertainties that can occur; performing risk-assessment in advance helps identifying such risks.

In this section we present several algorithmic approaches for finding efficient policies for planning under uncertainty and we argue that a mechanism design approach is preferable for all participating parties.

### 2.1 Centralised Planning

If we assume that the agent activity sets and cost functions are common knowledge (or agents are willing to disclose this information to a center), we are able to devise a centralised algorithm for developing efficient policies. We can combine all the individual agent MDPs and solve the joint MDP using a state-of-the-art MDP solver (e.g. Spudd [7]) to obtain the jointly-efficient policy.

Although this approach is most efficient in terms of total reward, it has several drawbacks. First of all, the assumption that agents are willing to disclose their private information in a competitive setting is rather strong. Secondly, this approach requires the agents to accept plans dictated by a center that might be optimal from a joint perspective but (a lot) more expensive for the agent himself.

A third practical issue is that finding efficient policies quickly becomes rather impractical from a computational point of view. We underline that this heavily depends on the encoding of the problem; in our experiments with the Spudd solver, the run time quickly becomes prohibitively large.

### 2.2 Individual Planning

The opposite of centralised planning is individual planning, in which agents develop their policies locally. This approach does not suffer from the autonomy and privacy issues as much (although the planner can derive information about the agent from its policy) and is computationally rather easy to solve. The drawback is of course that dependencies between agents and the effect of the policies on the user cost are completely ignored. Especially when charging the agents the part of the user cost they introduce, individual planning performs rather bad in terms of agent profit.

### 2.3 Best-Response Planning

Best-response planning (BRP, Jonsson [8]) can be considered a compromise between centralised and individual planning. Agents develop an initial joint plan, e.g. individually, and iteratively improve

on this plan in a Nash-like best-response way. In this way, the privacy and autonomy issues are minimised. In addition, computing the best response given a joint plan is far less computationally complex while we are still able to account for the impact of the plan on other agents and the network user (by incorporating this in the cost of a move). Note however that the BRP method depends on a given joint plan and can therefore not be easily used to find policies. Instead, we have to run the BRP algorithm in each time step, taking several iterations in each step to reach ‘reasonable’ solutions.

Another weakness of the BRP method is that, as we are using Nash dynamics to find joint plans, we have to settle for Nash equilibria in our solutions. This solution concept has not been widely studied for planning under uncertainty, hence we are unsure about the quality or even existence of such equilibria. In addition, the equilibria that can be reached by the BRP algorithm depends greatly on the order of moves. The BRP method can get trapped in local optima, preventing further iterations to reach (near-)optimal solutions.

## 2.4 Mechanism Design

As we are dealing with selfish, autonomous and rational agents, a more natural approach to solving our problem is to model the problem using *Algorithmic Game Theory (AGT)*.<sup>1</sup> Instead of a centralised solution in which all competitive agents need to reveal their private information, we design an incentive mechanism that aligns the individual objectives with the global goal of finding efficient joint plans. In this way it is in the best interest of players to plan efficiently, assuming they are rational.

This method overcomes the privacy and autonomy issues while still being able to produce optimal joint plans. Note that the computational aspect depends on the outcome rule and its corresponding algorithm of the mechanism that is being employed.

Nonetheless we cannot directly apply traditional mechanism design on planning problems with uncertainty for the reason stated in the introduction of this section. A ‘static’ mechanism is only able to produce a single plan and is therefore not robust against contingencies. Instead we focus on mechanisms in a dynamic setting. In the next section we present the theoretical background of *dynamic mechanism design*, required for the understanding of the novel dynamic planning mechanism we propose in Section 4.

## 3 Dynamic Mechanism Design

Dynamic mechanisms [1, 2, 3] extend ‘static’ mechanisms to deal with sequential games in which the private information of players evolves over time. In each time step, players<sup>2</sup> need to determine the best action (in expectation) to take while considering current private information and possible future outcomes. Many of the concepts defined in mechanism design have dynamic equivalents, which we present in this section. Most of the definitions in here originate from the work by Cavallo in [3].

### Definition 3.1: Direct Dynamic Mechanism

A dynamic mechanism  $M$  is a tuple  $\langle \pi, \mathcal{T} \rangle$  in which  $\pi : \Theta \rightarrow \mathcal{A}$  is a decision policy that maps the joint type space to the set of all possible actions<sup>3</sup>  $\mathcal{A}$  and  $\mathcal{T} = \{\mathcal{T}_i \mid i \in N\}$  is a set of payment functions  $\mathcal{T}_i : \Theta \rightarrow \mathbb{R}$ .

An important notion in mechanism design is the *revelation principle*[6]. This theorem states that any truthful indirect mechanism, in which agents only report outcomes, can be transformed into an incentive compatible direct mechanism, where agents disclose their full private information,

<sup>1</sup>For an introduction or refresher on AGT we refer the reader to for instance [11]. A short but excellent summary of the most relevant notions in the light of our research can be found in e.g. [4], although there the term computational mechanism design is used.

<sup>2</sup>We use the terms agent and player interchangeably. The term players is commonly used in game theory literature.

<sup>3</sup>The definition of this set depends on the encoding used. For instance, in a naive encoding for the centralised planning MDP this could be the Cartesian product of all agent activity sets.

and hence it suffices to study only the latter type of mechanism. An extension of the revelation principle for dynamic mechanisms also exists, due to the work by Pavan, Segal and Toikka [12].

The payment functions  $\mathcal{T}_i$  defined by the mechanism map a single joint type to a payment. In addition we define the expected payment functions  $\mathcal{T}_i^*$  that represent the expected payments from some joint type  $\theta^t$  until the end of the finite, discrete planning period  $T$ , given the mechanism policy  $\pi$  and the strategies  $\sigma$  (from the set of possible strategies  $\Sigma$ ) played by all the players. This function is given by  $\mathcal{T}_i^*(\theta_i^t, \pi, \sigma) = \mathbb{E}[\sum_{k=t}^T \gamma^{k-t} \mathcal{T}_i(\sigma(\theta^k))]$  given some discount factor  $\gamma \in [0, 1]$ .

The strongest solution concept in ‘static’ mechanism design, the dominant strategy equilibrium, is more difficult to realise in dynamic mechanisms, due to the contingent future. Instead we focus on equilibrium strategies that, when all agents adhere to them, in expectation yield maximum utility for every possible joint type  $\Theta = \theta_1 \times \dots \times \theta_N$ . This is defined as the within-period ex post Nash equilibrium:

**Definition 3.2:** Within-period ex post Nash equilibrium

*A strategy  $\sigma$  for a dynamic mechanism  $M$  constitutes a within-period ex post Nash equilibrium if and only if at all times  $t \in T$  for all players  $i \in N$ , all true types  $\theta^t \in \Theta$  and all other strategies  $\sigma' \in \Sigma$ :  $V_i(\theta^t, \pi, (\sigma_i, \sigma_{-i})) + \mathcal{T}_i^*(\theta^t, \pi, (\sigma_i, \sigma_{-i})) \geq V_i(\theta^t, \pi, (\sigma'_i, \sigma_{-i})) + \mathcal{T}_i^*(\theta^t, \pi, (\sigma'_i, \sigma_{-i}))$ . Here  $V_i$  denotes the (expected) value player  $i$  has for executing strategy  $\sigma_i$  given the current joint type  $\theta^t$  and the outcome chosen by  $\pi$  when others perform strategies  $\sigma_{-i}$ .*

Continuing along these lines, the notion of incentive compatibility can be extended to within-period ex post Nash equilibria. A mechanism is incentive compatible in this solution concept if for every joint type truthfully reporting maximises utility when other agents are also truthful. Mechanisms are individually rational in this equilibrium if for all joint types each agent is expected to make non-negative profit, given that all agents play a within-period ex post Nash equilibrium strategy. Finally, the mechanism is no-deficit if the sum of its payments to agents over the entire period  $T$  is non-positive. For the formal definitions of these concepts see [3].<sup>4</sup>

In [2], Bergemann and Valimaki proposed a dynamic version of the well-known VCG mechanism for the within-period ex post Nash equilibrium solution concept. Cavallo adapted this definition to fit in the dynamic mechanism framework in [3]. The dynamic-VCG mechanism (definition 3.3) yields maximum revenue among all mechanisms that satisfy efficiency, incentive compatibility and individual rationality in within-period ex post Nash equilibrium. Therefore we only restrict our focus to this mechanism in the rest of the paper.

**Definition 3.3:** Dynamic-VCG

*A direct, dynamic mechanism  $M = \langle \pi, \mathcal{T} \rangle$  belongs to the class of dynamic-VCG mechanisms if it satisfies:*

- (i) *The decision policy  $\pi$  is efficient for all time steps  $t$ , or  $\forall \theta^t \in \Theta$ :  $\pi = \arg \max_{\pi \in \Pi} \sum_{i \in N} V_i(\theta^t, \pi)$ .*
- (ii) *In every time step  $t$  players pay related to the additional cost they incur on other players, or  $\forall \theta^t \in \Theta, i \in N, \sigma_i \in \Sigma$ :  $\mathcal{T}_i^*(\theta^t, \pi, \sigma_i) = V_{-i}(\theta^t, \pi, \sigma_i) - C_i(\theta^t, \pi, \sigma_i)$  in which  $C_i : \Theta \rightarrow \mathbb{R}$  is an arbitrary function and  $C_i(\theta^t, \sigma_i) = \mathbb{E}[\sum_{k=t}^T \gamma^{k-t} C_i((\sigma_i(\theta_i^k), \theta_{-i}^k))]$ , i.e. the expected (discounted) value of the function  $C_i$ .*
- (iii) *Each player  $i$  pays exactly the expected valuation that other agents could have obtained forward from the joint type that follows from executing  $\pi$  without  $i$  participating, or  $\forall \theta^t \in \Theta, i \in N, \sigma_i \in \Sigma$ :  $\mathcal{T}_i^*(\theta^t, \pi, \sigma_i) = V_{-i}(\theta^t, \pi, \sigma_i) - V_{-i}(\theta_{-i}^t, \pi, \sigma_i)$*

Note that the first two conditions of definition 3.3 ensure that the mechanism is a member of the dynamic-Groves class [3]. The third condition is the dynamic version of the Clarke tax, similar to the traditional VCG mechanism.

<sup>4</sup>A weaker solution concept, the Bayes-Nash equilibrium is also defined for dynamic mechanisms, however we will not discuss this here as we are only interested in within-period ex post Nash equilibria. We refer the interested reader to [1] or [3].

## 4 Case Study: Infrastructural Networks

In this section we propose our dynamic mechanism for planning under uncertainty, using maintenance planning for infrastructural networks as a case study to aid the reader in understanding our method. This case study is motivated by the larger research project ‘Dynamic Contacting in Infrastructures’ in the context of which this work is being performed.

The example in this section is rather concrete, while our method is applicable on a large scale of planning problems. In general, our mechanism can be applied on any planning problem for which we want to find *efficient* plans in contingent environments where the participating agents are willing to disclose their private information. By using an indirect mechanism, this requirement can be relaxed but this is not in the scope of this paper.

### 4.1 Infrastructural Networks

The problem we study in this section arises in maintenance of public infrastructures such as e.g. road networks, power grids or the internet. Commonly there is a public asset manager (i.e. governmental institution, public organisation, etc.) responsible for the network quality at the behalf of the (tax-)paying user. Moreover, there is a cost associated with the use of the network, for instance travel time in road networks or packet travel times.

An infrastructural network is defined by  $\mathcal{N} = \langle G, \ell, Q \rangle$ .  $G = \langle V, E \rangle$  is a representation of the network topology. With each edge  $e \in E$  we associate a quality level  $q_e$ .<sup>5</sup> The cost to the user for a flow (e.g. traffic, packets, etc.)  $f$  on the network is captured by the function  $\ell(f) \in \mathbb{R}$ . In addition, we define the *marginal user cost* of any flow  $f$  in respect to the flow  $f^0$  that occurs when all edges are completely available as  $\ell^*(f) = \ell(f) - \ell(f^0)$ .

Finally,  $Q : q \rightarrow \mathbb{R}$ , where  $q = \bigcup_{e \in E} q_e$ , is a function that maps quality levels  $q_e$  to a cost in order to express the monetary impact of the edge quality states. This cost is imposed by the asset manager on the service provider responsible for the edge to incentivise higher quality levels.

### 4.2 Maintenance Planning

In our problem we are dealing with infrastructural networks that are not serviced by the asset manager himself. Instead, maintenance is done by a set of  $N$  service providers (agents), each having a unique set of maintenance activities<sup>6</sup>  $A_i$  at their disposal, requiring unit time to perform. Each task  $a \in A_i$  is defined by the tuple  $\langle e, q', \alpha \rangle$ , containing the edge  $e$  that is serviced, the effect  $q'$  on the quality and a success rate  $\alpha \in [0, 1]$ . In addition, agents have a private cost function  $c_i : A_i \times T \rightarrow \mathbb{R}$  that for each of their tasks captures the cost of performing it at any time step.

Given their activities and cost function, the goal of the agents is to develop a maintenance plan  $P_i$ . This plan is a sequence of tasks  $\{a_0, a_1, \dots, a_{|T|-1}\}$  chosen from  $A_i \cup \{\circ\}$ , with  $\circ$  being a no-operation task. The cost of such a plan for the agent is simply the sum of task costs, or  $C_i(P_i) = \sum_{t \in T} c_i(P_i(t), t)$ .

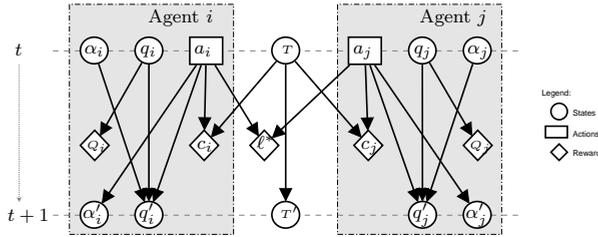
We are interested in finding jointly *efficient*<sup>7</sup> plans that minimise the total cost of performing maintenance on the infrastructural network – i.e. task costs, quality costs and the (increase in) user costs – and not just the agent’s individual cost. Simply solving these individual planning problems and combining them to form the joint plan  $P = \bigcup_{i \in N} P_i$  neglects both the network quality as well as the impact the joint maintenance plan has on the network user. Denoting the task cost of the joint plan by  $C(P) = \sum_{i \in N} C_i(P_i)$ , then we are interested in the solutions:  $P^* = \arg \min_{P \in \mathcal{P}} C(P) + \sum_{t \in T} (Q(q^t) + \ell^*(f^t(P)))$  in which  $q^t$  is the vector of edge qualities updated according to the chosen activities (and their success),  $f^t(P)$  the flow resulting from the planned tasks and  $\mathcal{P}$  the set of all possible plans.

However, as discussed before, finding a single joint plan does not guarantee efficiency in contingent environments. Each task has a success rate  $\alpha \in [0, 1]$  such that they can fail with some

<sup>5</sup>The model does not depend on the actual choice for quality level representation, we could use for instance discrete levels, a normalised relative factor  $[0, 1]$ , etc.

<sup>6</sup>We use the terms activities and tasks interchangeably.

<sup>7</sup>The term efficient solution is commonly used in game theory for maximising overall utility, similar to our goal of minimising maintenance cost for all players.



**Figure 1** – A two-player example that represents two time steps of the MDP formulation for the maintenance planning problem.

random probability  $1 - \alpha$  and we need to react to such possible failures. Therefore we apply a dynamic mechanism, presented in the next sub section.

Note that other sources of uncertainty could be the task duration or the actual quality of an edge, but this simple modification allows us to study the complex dynamics of a uncertain world, while keeping the model conceptually easy .

### 4.3 Efficient Maintenance Planning Mechanism

Before we can apply a mechanism, we first need to formulate the planning problem as a game. The players (service providers) are modelled as rational, self-interested and autonomous agents that want to maximise their own utility. Each of the agents have a type space  $\Theta_i$  that defines its local set of MDP states  $S_i$ . From this they choose for each time step  $t$  a type  $\theta_i^t = (s_{\theta_i^t}, r_{\theta_i^t}, \tau_{\theta_i^t})$ , with  $s_{\theta_i^t}$  representing the current state,  $r$  the reward function and  $\tau$  the transition function as before.

Instead of presenting the complex MDP formulation, we illustrate our model with a two-agent example in Figure 1.

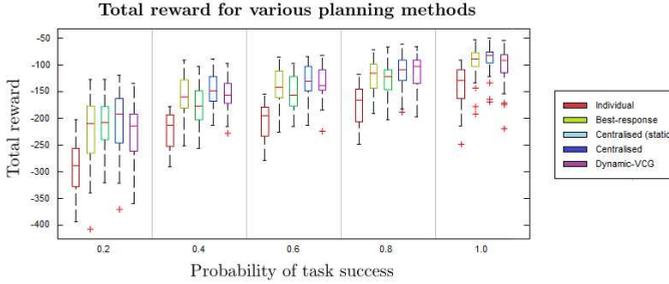
In this figure we have depicted the state variables, the chosen actions and their dependencies for two time steps.<sup>8</sup> Also, the MDP rewards for the choices made in time  $t$  are illustrated by the diamonds in between the two states.

Agent utilities in dynamic mechanism design do not only depend on the valuation of the current outcome but also on the (expected) valuation of future outcomes. In order to compute its utility, agents also need information about possible future outcomes and strategies played by others. This functions is defined as  $u_i(\theta^t, \pi, \sigma) = V_i(\theta_i^t, \pi, \sigma) - \mathcal{T}_i^*(\theta_i^t, \pi, \sigma)$  in which  $\pi$  is the policy decision function of the mechanism that maps types to outcomes,  $\sigma$  the set of strategies players use from the set of available strategies  $\Sigma$ ,  $V_i$  the player valuation function and  $\mathcal{T}_i^*$  the expected payment that this player has to pay to or receives from the center. Note that both the valuation and payment are not computed over a single type; instead, they are (possibly discounted) expectations over future outcomes given the policy and agent strategies. Agent valuations are defined as  $V_i(\theta^t, \pi, \sigma) = \mathbb{E}[\sum_{k=t}^T \gamma^{k-t} r_i(\theta_i^k, \pi(\sigma(\theta^k)))]$  given the discount factor  $\gamma \in [0, 1]$  and their MDP reward function  $r_i$ .

From Figure 1 we can see that each agent’s reward function consists of the quality cost  $Q_i$  and its private cost  $c_i$  component. The revenue for doing their work is not included as part of the reward. This is because we assume w.o.l.g. that they will receive a fixed, contracted price and therefore only the cost components have to be optimised.

In addition to the service providers – the players we discussed so far – we also include the asset manager as a player, denoted by  $\phi$ . The asset manager can be seen as the center of the mechanism, representing the network authority and acting on the behalf of the network users. The type space  $\Theta_\phi$  of the center contains the infrastructural network  $\mathcal{N}$  and each type  $\theta_\phi^t \in \Theta_\phi$  includes the network quality vector  $q^t$  and the current flow  $f^t$ . The asset manager cannot take actions, but its valuation function accounts for the network quality and user costs. Its value function  $V_\phi$  is given by  $V_\phi(\theta^t, \pi, \sigma) = \mathbb{E}[\sum_{k=t}^T \gamma^{k-t} \ell^*(\pi(\sigma(\theta^k)))]$ , thus the expected quality and user cost given the joint type that results from policy  $\pi$  when players employ strategies  $\sigma$ .

<sup>8</sup>Note that the notation is not entirely correct, however to keep the figure readable we subscript the action, quality and success rate with the corresponding player index.



**Figure 2** – Total plan rewards for various planning methods. With a 100% success rate, the static centralised and dynamic centralised algorithm are equivalent.

If we include the center in the set of players, i.e.  $N^* = N \cup \{\phi\}$ , then we can define a dynamic-VCG mechanism for the maintenance problem. At each time  $t$  players report their private information  $\theta_i^t$  to the center, who is able to develop efficient maintenance plans using the policy that can be computed using e.g. the centralised method presented in sub Section 2.1. I.e. the asset manager combines the local MDP's, finds the optimal policy and dictates the actions to be taken by the service providers.

## 5 Experiments

In order to compare and validate the dynamic mechanism we propose, we have implemented all of the algorithmic approaches presented in Section 2 and tested them on a set of 60 generated, two-player maintenance planning problem (Section 4) instances. These instances are generated using the following settings:

- Each player is responsible for one road only, with a discrete quality between 0 and 6. The quality cost function is linearly decreasing in the quality by some random factor from  $[1, 3]$ .
- Players have equi-sized task sets of size 1, 2 or 3 (excluding the no-op). Of each task set size we generated 20 instances. Tasks increase the quality level by 1, 2 or 3 units (corresponding to the task set sizes, i.e. a task set of size 2 contains a 1 and 2 unit effect task) and have a linear cost in the effect with a random factor from  $[1, 3 * \text{effect}]$ .
- The user cost is given by a penalty that is only awarded when both players perform an activity at the same time. The penalty value differs for the two players and is chosen randomly from an interval  $[1, 10]$ .

We ran all four algorithms on these 60 instances with different levels for the task success rates using a mixed-integer encoding. In addition, we also included the 'static' centralised algorithm that develops a plan in advance but does not respond to any task failures. Note that we used the same random seed for all methods so that the same contingencies occur.

### 5.1 Discussion

In the context of this paper we have chosen to include only the total plan reward results, depicted in Figure 2. From the efficiency perspective, the centralised algorithm can be considered the optimal method. The individual planner performs rather weak efficiency-wise as expected. The best-response method performs remarkably well (although taking a lot of time, already for the 3 iteration rounds we used) and is able to produce near-optimal plans.

Finally, we can see that the total rewards of our mechanism are close to optimal (i.e. compared to the centralised algorithm). Nonetheless, when no failures occur the mechanism produces rather expensive plans compared to the centralised method. With a low rate of success, the mechanism does not perform well. This is due to the fact that under high task failure rates, players have to make the same payments for the user cost they would cause to others again when rescheduling the task at a later time.

## 6 Conclusion

We presented several known algorithmic approaches and a novel dynamic mechanism for solving multi-agent planning under uncertainty. The main advantage of the mechanism is that under the mechanism players are compensated for their contribution towards efficient plans. Under the dynamic-VCG payments players have a higher utility for globally efficient plans than plans that minimise their own private cost. Our experiments support the application of such a mechanism to this domain.

Regretfully, the dynamic-VCG mechanism suffers a few weaknesses that may render it quite impractical for real-world application. Most notably the mechanism requires the participating agents (commercial service providers) to disclose all of their private information concerning maintenance activities and costs. In the competitive setting in which these agent operate, this might be unacceptable and therefore future research should be directed towards indirect dynamic mechanisms.

### Acknowledgements

This research is part of the Dynamic Contracting in Infrastructures project and is supported by Next Generation Infrastructures and Almende BV. Matthijs Spaan is funded by the FP7 Marie Curie Actions Individual Fellowship #275217 (FP7-PEOPLE-2010-IEF).

## References

- [1] S. Athey and I. Segal. An efficient dynamic mechanism. Technical report, UCLA Department of Economics, 2007.
- [2] D. Bergemann and J. Valimaki. Efficient dynamic auctions. *Cowles Foundation Discussion Papers*, 2006.
- [3] R. Cavallo. Efficiency and redistribution in dynamic mechanism design. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 220–229. ACM, 2008.
- [4] R.K. Dash, N.R. Jennings, and D.C. Parkes. Computational-mechanism design: A call to arms. *IEEE intelligent systems*, pages 40–47, 2003.
- [5] T. Dean, L.P. Kaelbling, J. Kirman, and A. Nicholson. Planning with deadlines in stochastic domains. In *Proceedings of the eleventh national conference on Artificial intelligence*, volume 574, page 579. Citeseer, 1993.
- [6] A. Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41(4):587–601, 1973.
- [7] J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier. Spudd: Stochastic planning using decision diagrams. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 279–288. Morgan Kaufmann Publishers Inc., 1999.
- [8] A. Jonsson and M. Rovatsos. Scaling up multiagent planning: A best-response approach. In *Twenty-First International Conference on Automated Planning and Scheduling*, 2011.
- [9] J.M. Moonen. *Multi-agent systems for transportation planning and coordination*. Erasmus University Rotterdam, 2009.
- [10] Y. Narahari, D. Garg, R. Narayanam, and H. Prakash. *Game theoretic problems in network economics and mechanism design solutions*. Springer-Verlag New York Inc, 2009.
- [11] N. Nisan. *Algorithmic game theory*. Cambridge Univ Pr, 2007.
- [12] A. Pavan, I. Segal, and J. Toikka. Dynamic mechanism design: Revenue equivalence, profit maximization and information disclosure. 2008.
- [13] S Seuken, R Cavallo, and D C Parkes. Partially-synchronized dec-mdps in dynamic mechanism design. In *Proceedings of the 23rd national conference on Artificial intelligence*, volume 1, pages 162–169, 2008.

# Multiple Instance Learning using Bag Distribution Parameters

D.M.J. Tax

*Delft University of Technology, Delft*

## Abstract

In pattern recognition and data analysis, objects or events are often represented by a feature vector with a fixed length. For some applications this is a severe limitation, and extensions have been proposed. One approach is Multiple-Instance Learning (MIL). Here, objects are represented by a *collection* of feature vectors (called a bag) and a bag is labeled positive, when at least one feature vector is member of a *concept*. In some situations it is not suitable to assume the presence of a concept, and the distribution of all the feature vectors in a bag is required to classify the bag. In this paper we propose a simple bag classification scheme using the parameters of the fitted distributions. Experiments show sometimes surprisingly good performances with respect to other state-of-the-art approaches.

## 1 Introduction

A standard assumption in pattern recognition or machine learning is, that objects are represented by a fixed number of features values. This reduces a real-world object or event to a single point in a feature space, and allows for a relatively straightforward analysis and classification of the object. Unfortunately, this assumption is often very restrictive. Consider, for instance, the representation of an image containing a scene with objects. In order to capture all information about all objects, a very rich representation is required. Only when the task is specified beforehand precisely (for instance, that we want to label images as ‘horses’ or ‘no-horses’) more specific segmentation and feature extraction can be defined. If the representation should be more flexible, to allow for more flexible class definitions, the single feature representation becomes hard.

A well-known approach to classify a set of instances, is to use a ‘bag of words’ approach [9]. Here a dictionary of words is defined, what should represent all possible concepts in the feature space. This is typically found by clustering all instances from a large collection of objects, and using the cluster centers as words. The instances in an object are assigned to the best matching word, and the object is represented by a word-presence vector. This approach has the advantage that the classification typically does not depend on a single word (like it depended on a single concept in MIL). But the drawback is that a suitable collection of words has to be defined (which can become hard when the total number of instances is not very large), and that the word-presence vector typically becomes very sparse.

An alternative approach is to break up the object in smaller, and more homogeneous, parts, and to represent the object by a collection of parts. This is called Multiple Instance Learning (MIL)[5]. Instead of a single feature vector, we have a collection (called a bag) with feature vectors (called instances). For the representation of images it would mean that the image is split in smaller patches or homogeneous segments, and each of these segments is characterized by standard image features like color, texture and shape features. An example is shown in the first three subfigures in Figure 1. An original image containing a horse (subfigure 1(a)) is segmented in non-overlapping segments (subfigure 1(b)), where from each of the segments a feature vector is extracted. The classifier then classifies the individual segments, and applies a combining rule to derive a bag label.

In MIL typically an additional assumption is made; the task is to detect the presence of a so-called concept. Therefore bags have to be classified to a positive or negative class (like ‘horses’ and ‘no-horses’). A bag is positive, when at least one instance in this bag is member of the concept, and otherwise the bag is labeled negative. For the example in Figure 1 the segments that belong to the concept ‘horse’ are shown in

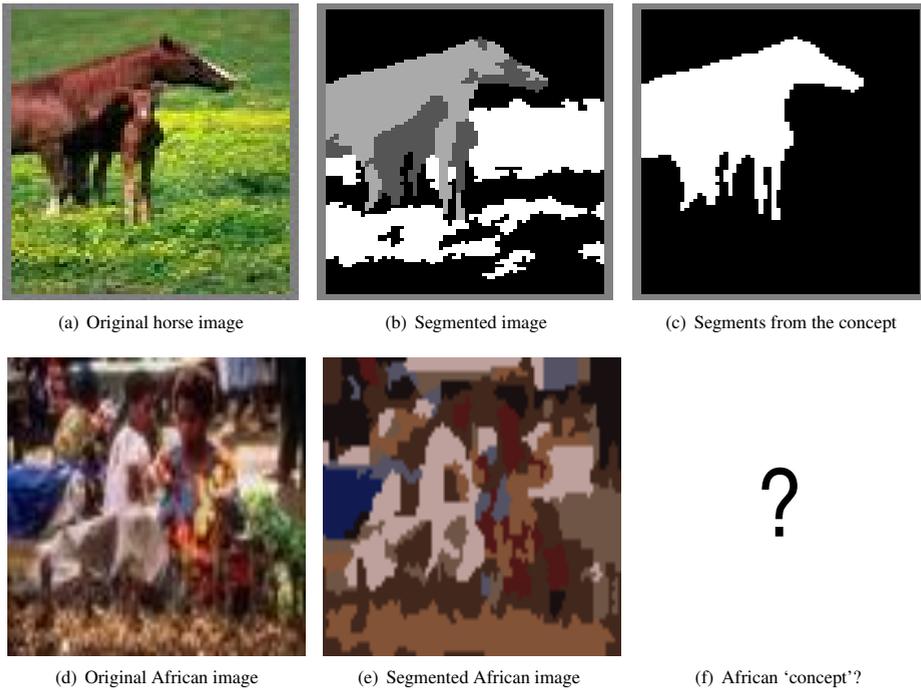


Figure 1: Images can be segmented in homogeneous segments, and are thus represented by a collection of image segments. For image classification problems that involve clear objects, some of these segments should belong to the concept. For more abstract classification problems, like the class ‘African’ for the image in subfigure 1(d), it may be hard to find such concept. Instead, the distribution of all image segments has to be considered.

white in subfigure 1(c). To find a good classifier, we face two problems: what instances are informative for the class label (or, are member of the concept), and how can we train a good classifier to model this concept?

A drawback of this concept-modeling approach is that it essentially assumes that these concepts are also present as tight clusters in the feature space. ‘Vague’ and higher level concepts are therefore not suitable for this approach. For instance, when images have to be labeled ‘outdoor scene’, ‘evening’ or ‘African’. Here it can be expected that most instances in the bag are informative, and not a single one as is assumed in the strict MIL definition. This is shown in the second row of figures in Figure 1, where an image labeled ‘African’ is segmented, but no clear concept can be distinguished. For these types of problems, it may be more natural to characterize the complete distribution of instances in a bag, instead of just selecting a single, most informative, one.

In this paper we propose an alternative representation, that avoids the pre-definition of concepts or words. Instead, the bag of instances is modeled by a full probability distribution. That means that bag distributions have to be classified. In [7] kernels between bags have been defined, and on these kernels a support vector classifier is trained. Here we propose a simpler alternative to classify the bag distribution: fit a probability density model per bag, and use the optimized parameter values as the feature representation of the bag. This is typically very simple to perform, and allows us to use any classifier for the classification. A drawback is that the number of instances per bag is typically limited, and therefore only simple density models can be used. In the experimental section we show that for many problems the performances are still surprisingly good (i.e. easily beating more advanced MIL approaches).

In section 2 we discuss a few possible density models and their feature representations. In section 3 some experiments with comparisons show the feasibility of the approach and in 4 we conclude.

## 2 Modeling bags with distributions

Assume that we have a training set with labeled bags  $\mathcal{X} = \{(b_i, y_i); i = 1, \dots, N\}$ , where each of the bags contains  $n_i$  instances  $b_i = \{\mathbf{x}_i; i = 1, \dots, n_i\}$  and the instances are standard feature vectors  $\mathbf{x} \in \mathbb{R}^d$ . In standard MIL it is assumed that there is a positive and negative class,  $y = +1$  or  $y = -1$ , and that the bag label is derived from the instance labels. The instance are labeled according to their membership of a so-called concept: when instance  $\mathbf{x}$  belongs to the concept  $c(\mathbf{x}) = +1$ , otherwise  $c(\mathbf{x}) = 0$ . The bag label is then inferred using:

$$\hat{y}(b_i) = \begin{cases} +1 & \exists k : c(\mathbf{x}_{ik}) = +1 \\ -1 & \text{otherwise.} \end{cases} \quad (1)$$

The typical approach to train a MIL classifier, is to optimize an instance classifier  $h$  that classifies each instance  $\mathbf{x}$  to be concept or not. Then the combining rule (1) is applied to obtain the bag label.

For the general, optimal bag classification, the class posterior probability given all the instances in a bag ( $p(y|b)$ ) has to be found:

$$\hat{y} = \arg \max_y p(y|b_i) = \arg \max_y p(y|\{\mathbf{x}_{i1}, \dots, \mathbf{x}_{in_i}\}) \quad (2)$$

This model is very flexible, because *all* instances or all combinations of instances in a bag can potentially contribute to the discrimination between classes. This model family is too broad to handle easily, and therefore here we make the simplifying assumption that the *distribution* of the instances is informative for the class label.

When we model, or approximate, the collection of instances by a density  $\hat{d}$ , the class posterior probability is now estimated by:

$$p(y|b_i) = p(y|\hat{d}(\mathbf{x}|\boldsymbol{\theta}_i)) = p(y|\boldsymbol{\theta}_i) \quad (3)$$

where  $\boldsymbol{\theta}_i$  are the parameters of the density  $\hat{d}$  what was fitted on bag  $b_i$ . Note that we avoid the explicit representation of  $\hat{d}$  in the feature space, but instead use only the density parameters. It is therefore assumed that all information in the collection of instances is captured by the distribution parameters  $\boldsymbol{\theta}$ . This is indeed true, when the instances are truly drawn from the distribution  $\hat{d}$ , and the parameters  $\boldsymbol{\theta}$  are the sufficient statistics of  $\hat{d}$ . Typically, the parameters  $\boldsymbol{\theta}_i$  are then estimated using maximum likelihood on the given bag instances  $\{\mathbf{x}_{i1}, \dots, \mathbf{x}_{in_i}\}$ .

In general, when the bag distribution model is flexible enough to characterize the distribution of the instance well, the distribution parameters should be informative enough for the subsequent classifier. To estimate the class posteriors  $p(y|\boldsymbol{\theta})$ , or to perform the final classification (2), we can apply any ‘regular’ classifier on the parameters  $\boldsymbol{\theta}$ . For instance a logistic classifier, an LDA, a support vector classifier or any other density based classifier can be fitted. Therefore, instead of explicitly estimating and modeling a concept, the classification problem is represented in terms of the distribution parameters of each of the individual bags.

### 2.1 Bag density models

Typically, the number of instances per bag is limited, so it is hard to estimate very complicated density models on each of the bags. Furthermore, the number of parameters has to be equal for all bags  $b_i$ , therefore nonparametric distributions cannot be used. The remaining possibilities are still numerous though. Below we list a few simple approaches:

**Gaussian distribution** The bags are characterized by both a mean and a covariance matrix:

$$p(y|b_i) = p(y|[\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i]) \quad (4)$$

where the brackets  $[\mathbf{a}]$  means that all values of  $\mathbf{a}$  are concatenated into one vector. The  $\hat{\boldsymbol{\mu}}_i$  and  $\hat{\boldsymbol{\Sigma}}_i$  are the maximum likelihood estimates of the mean and covariance matrix. For a  $d$ -dimensional Gaussian, this results in  $d + d(d+1)/2 = (d^2 + 3d)/2$  dimensional bag representation.

**Gaussian with identical covariance matrices** When equal covariance matrices are assumed for all bags, the parameters of these covariance matrices can not be informative for discriminating classes. The only parameter left is the mean vectors per bag:

$$p(y|b_i) = p(y|\hat{\boldsymbol{\mu}}_i) \quad (5)$$

This results in a representation with  $d$  feature values for each bag.

**Uniform distribution** The maximum likelihood solution for a (axis parallel) uniform distribution is the rectangle tightly fit around the instances. These boundaries are the minimum and maximum feature values that appear somewhere in the instances of a bag. So when we define the minimum

$$\mathbf{b}_i^l = [\min_j((\mathbf{x}_{ij})_1), \dots, \min_j((\mathbf{x}_{ij})_d)] \quad (6)$$

(where  $(\mathbf{a})_k$  indicates the  $k$ -th feature of vector  $\mathbf{a}$ ) and maximum

$$\mathbf{b}_i^u = [\max_j((\mathbf{x}_{ij})_1), \dots, \max_j((\mathbf{x}_{ij})_d)] \quad (7)$$

we obtain:

$$p(y|b_i) = p(y|[\mathbf{b}_i^l, \mathbf{b}_i^u]) \quad (8)$$

The resulting bag feature vector has now  $2d$  components.

An alternative encoding for the same parameters can be made by using the center and the width of the uniform distribution:

$$p(y|b_i) = p(y|[(\mathbf{b}_i^u + \mathbf{b}_i^l)/2, (\mathbf{b}_i^u - \mathbf{b}_i^l)/2]) \quad (9)$$

**Poisson distribution** When the features are counts (like word-counts in a document characterization), a Poisson distribution per feature can be estimated. The maximum likelihood for the rates  $\lambda_i$  are the average of the counts, therefore this estimator reduces to (5).

Note that the final practical implementation of this idea is very similar to the approach in [7]. In that paper it was noted that it is possible to extract features from bags of instances, and use these as input to a standard classifier. That is basically what is being done here, except that here the features directly follow from the assumption a user makes on the distribution of the instances in the bags. So when some knowledge on the bag distributions may be available, it may give a hint what features can be used.

## 2.2 Alternative approaches

In the original MIL formulation, the basic assumption is that there is a concept. Many standard MIL approaches therefore search for the most informative instances in a bag, and try to optimize a concept model on these instances. In the seminal work [5] an axis parallel rectangle (APR) is fitted, while later in [10] a probabilistic formulation is given, called Diverse Density (here often a axis parallel Gaussian model for the concept is used). Unfortunately, these methods tend to be very computational intensive, because it does not just involve optimizing a concept model, it also includes the selection of the most suitable instances from the positive bags.

Therefore many approximate formulations are proposed. For instance, in [2] MISVM is explained. Here a support vector classifier is trained by selecting just a single instance from each positive bag as a ‘witness’. All instances from the negative bags are treated as negative examples. The selection of the positive instances can be iterated once, or a few times. In [12] MiBoost is defined, that performs a boosting where all instances get a weight, according to their apparent importance for the classification task.

When a concept can be identified in the classification problem, these approaches tend to work well. On the other hand, in some situations it appears that the identification of a concept is very hard. A naive, but still successful, approach is to copy all the bag labels to the instance labels, and train a standard classifier on the instances [13]. To classify a new and unseen bag of instances, all instances are classified first and a standard combining rule is used to fuse the labels. Although this approach ignores the bag structure in the data, it can obtain good results.

Finally, the standard approach in computer vision classification tasks is to use a ‘bag of words’ approach [6] (which originated from natural language processing). Here, all the instances are clustered (or, like in [4], all instances are used) and each of the clusters is considered a potential concept. The bags are now represented by the similarity to all the words. This results in a feature representation of fixed length, that can be classified by a standard classifier.

### 3 Experiments

To compare the basic bag representation with alternative MIL approaches, experiments are performed on a large set of datasets. Many MIL problems are image classification problems, because images can be naturally represented by a bag of instances. First, images are segmented, then features per segment are extracted and different classes are defined [4, 2, 3]. Three of these problems are considered in this paper. Some other types of problems are considered as well. First the classical drug discovery problems, Musk1 and Musk2, in which molecules are described by 166 shape features [5] and have to be distinguished between active and inactive molecules. Another dataset considers newsgroup classification, in which newsgroup articles are described by 200 TFIDF features [15] and given a collection of newsgroup articles, the newsgroup name has to be predicted. And finally web page preference prediction, where web pages are described by web pages that link to the page of interest [14].

Table 1: Dataset characteristics of some standard MIL datasets.

dataset	nr.inst.	dim.	pos. bags	neg. bags	min. inst/bag	median inst/bag	max. inst/bag
MUSK 1 [5]	476	166	47	45	2	4	40
MUSK 2 [5]	6598	166	39	63	1	12	1044
Corel African [4]	7947	9	100	1900	2	3	13
Corel Historical [4]	7947	9	100	1900	2	3	13
SIVAL AjaxOrange [8]	47414	30	60	1440	31	32	32
News atheism [15]	5443	200	50	50	22	58	76
Corel Fox [1]	1320	230	100	100	2	6	13
Corel Tiger [1]	1220	230	100	100	1	6	13
News motorcycles [15]	4730	200	50	50	22	49	73
News mideast [15]	3373	200	50	50	15	34	55
Harddrive [11]	68411	61	178	191	2	290	299
Web recomm.[14]	2212	5863	17	58	4	24	141

In table 1 some characteristics are shown of the datasets that are considered in this paper. The datasets are chosen to show some variability in number of features, number of bags, and (average) number of instances per bag.

Table 2: AUC performances ( $100\times$ ) of the classifiers on datasets Musk1, Musk2, Corel African and Corel Historical. Results are obtained using five times 10-fold stratified crossvalidation.

classifier	Musk 1	Musk 2	Corel African	Corel Historical
APR	78.9 (1.7)	80.8 (2.3)	57.4 (0.8)	61.4 (0.4)
Diverse Density	<b>90.3 (1.8)</b>	<b>93.2 (0.0)</b>	85.6 (0.1)	54.4 (0.2)
Boosting MIL	80.3 (3.1)	49.3 (3.7)	64.1 (0.1)	38.0 (0.4)
Citation kNN	88.6 (2.1)	82.9 (1.2)	80.4 (1.6)	76.5 (0.9)
MI-SVM	70.3 (3.0)	81.5 (2.1)	63.5 (1.7)	78.8 (1.5)
MILES	89.3 (1.9)	88.8 (1.8)	88.5 (0.5)	<b>90.8 (0.8)</b>
Naive MIL with Logistic	77.8 (1.7)	80.5 (1.8)	75.7 (0.2)	82.8 (0.1)
BagOfWords (k=10) Logistic	66.2 (5.6)	65.5 (5.9)	76.0 (2.4)	81.5 (2.0)
BagOfWords (k=100) Logistic	80.9 (4.7)	74.8 (4.1)	83.4 (2.3)	86.1 (2.5)
Mean-inst Logistic	84.3 (1.2)	89.6 (2.4)	83.2 (0.3)	88.6 (0.2)
Extremes Logistic	89.9 (2.5)	89.2 (2.3)	<b>90.0 (0.2)</b>	86.5 (0.2)
Extremes2 Logistic	<b>91.8 (2.0)</b>	90.0 (1.8)	<b>90.0 (0.2)</b>	86.6 (0.2)
Covariance Logistic	78.0 (3.5)	<sup>(a)</sup>	86.5 (0.9)	88.5 (0.3)

On these datasets the MIL classifiers mentioned in this paper are trained. The Axis Parallel Rectangle (APR) and the Diverse Density explicitly model a concept, while the Boosting MIL, the MI-SVM and MILES try to select the most informative instances, without modeling them. Next to that, the naive MIL implementation is used [13], a bag of word representation is used, and this is compared to the representa-

tions that are derived in section 2.1. Note that for the uniform bag distribution, two alternative parameter formulations are defined. They are called ‘Extremes’ (equation (6)) and ‘Extremes2’ (equation (8)). The ‘Mean-inst’ entry shows the results of the Gaussian with identical cov. matrices, equation (5). For the approaches that define representations, but not classifiers, the (linear) logistic classifier is used.

In Tables 2, 3 and 4 the Area Under the ROC curve performances of the MIL classifiers are shown for the datasets mentioned in Table 1. There are some situations that experiments cannot be performed. In situations <sup>(a)</sup> where a bag contains just a single instance, it is not possible to get a good estimate of the covariance matrix, and no results are given. Furthermore, in <sup>(b)</sup> for very large datasets (like AjaxOrange and Harddrive), the MILES optimizer runs out of memory. Similarly, in <sup>(c)</sup> the dimensionality of the dataset is so large, that it becomes infeasible to estimate a covariance matrix.

In the text-categorization problems ‘News atheism’, ‘News motorcycles’ and ‘News mideast’ the feature vectors are very sparse, resulting in very small distances between the vectors. Because most instances are now very close together, it becomes very hard to find 100 distinct clusters. Results for 100 words could therefore not be reliably obtained. This is indicated by <sup>(d)</sup>. Finally, the optimization of the Diverse Density is slow, and for the large Harddrive dataset one fold in the ten-fold crossvalidation experiments required more than 20 hours, therefore no result is obtained <sup>(e)</sup>.

The first results listed in Table 2 already show that in particular the ‘Extremes’ representation is very suitable for the Musk datasets, it approaches the much more advanced Diverse Density approach. Also for the Corel African (from which the example images in Figure 1 are taken), this performs very well. The approaches that try to select informative instances, perform worse here. For Corel Historical the mean instance representation is suitable, although it does not outperform the MILES classifier.

Table 3: AUC performances (100×) of the classifiers on datasets AjaxOrange, News atheism, Fox and Tiger. Results are obtained using five times 10-fold stratified crossvalidation.

classifier	AjaxOrange	News atheism	Fox	Tiger
APR	48.4 (0.8)	50.0 (0.0)	55.2 (1.2)	57.9 (1.6)
Diverse Density	55.5 (0.2)	42.0 (0.0)	<b>66.8 (1.2)</b>	80.8 (1.5)
Boosting MIL	NaN (0.0)	50.0 (0.0)	53.5 (1.4)	74.2 (1.3)
MI-SVM	93.6 (2.6)	69.8 (2.8)	54.4 (1.5)	80.1 (1.1)
MILES	<sup>(b)</sup>	80.4 (1.2)	61.6 (0.9)	<b>82.5 (1.1)</b>
Naive MIL with Logistic	95.2 (0.1)	82.7 (2.5)	54.9 (3.2)	80.0 (1.4)
BagOfWords (k=10) Logistic	66.4 (2.7)	63.4 (8.5)	56.8 (3.5)	71.2 (4.6)
BagOfWords (k=100) Logistic	77.9 (1.7)	<sup>(d)</sup>	53.7 (4.9)	63.1 (5.2)
Mean-inst Logistic	87.5 (0.6)	<b>85.2 (2.2)</b>	55.4 (2.2)	79.5 (2.0)
Extremes Logistic	91.6 (0.4)	81.8 (3.4)	60.7 (2.8)	77.5 (2.0)
Extremes2 Logistic	91.6 (0.4)	81.0 (3.0)	60.1 (2.1)	76.6 (2.8)
Covariance Logistic	<b>97.7 (0.8)</b>	75.3 (2.0)	62.7 (1.9)	<sup>(a)</sup>

Often the characterization of a bag of instances using the extreme feature values performs very well. It can even approach the performance for concept-based MIL classifiers for datasets that appear to have a concept present. This could be explained when we assume that the concept is ‘sticking out’ in one of the directions in the feature space. If this direction is parallel to a few feature dimensions, it automatically will result in an extreme value for the minimum or maximum value for one of the features that characterizes the positive bags. These extreme values can therefore be considered as a poor-man’s concept description.

In Tables 3 and 4 also the mean-instance representation (using (5)), or the representation using both the mean and covariance matrix (equation (4)) perform surprisingly well. In particular for the text-based problems, like the web recommendation and the News problems, the mean-instance representation is the most suitable representation.

For the datasets Fox and Tiger, a clear concept appears to be present in the objects, and selecting the most informative instances becomes essential. Taking into account all instances in a bag confuses and adds too much noise to the representation, and deteriorates the performance. For these problems it is required to apply Diverse Density or MILES.

Table 4: AUC performances ( $100\times$ ) of the classifiers on datasets News motorcycles, News mideast, Harddrive and Web recommendation. Results are obtained using five times 10-fold stratified crossvalidation.

classifier	News motorcycles	News mideast	Harddrive	Web recomm.
APR	50.0 (0.0)	49.8 (0.4)	90.4 (2.3)	58.9 (3.6)
Diverse Density	46.4 (2.9)	40.2 (2.5)	<sup>(e)</sup>	NaN (0.0)
Boosting MIL	50.0 (0.0)	62.1 (4.2)	44.4 (0.1)	74.1 (4.6)
MI-SVM	76.4 (4.0)	<b>79.8 (2.3)</b>	NaN (0.0)	60.7 (4.3)
MILES	77.4 (1.9)	<b>79.9 (3.4)</b>	<sup>(b)</sup>	70.5 (5.0)
Naive MIL with Logistic	76.8 (3.4)	<b>80.4 (2.2)</b>	91.9 (0.7)	76.7 (1.8)
BagOfWords (k=10) Logistic	64.6 (6.0)	68.2 (5.2)	<b>96.6 (1.7)</b>	50.0 (0.0)
BagOfWords (k=100) Logistic	<sup>(d)</sup>	<sup>(d)</sup>	<b>96.8 (1.0)</b>	50.0 (0.0)
Mean-inst Logistic	<b>85.3 (0.7)</b>	<b>79.4 (3.2)</b>	<b>96.0 (0.6)</b>	<b>90.0 (0.4)</b>
Extremes Logistic	77.8 (2.1)	<b>76.8 (2.5)</b>	94.4 (0.5)	81.5 (0.8)
Extremes2 Logistic	78.5 (1.7)	<b>76.8 (2.5)</b>	94.2 (1.0)	81.3 (0.7)
Covariance Logistic	74.5 (2.7)	73.1 (2.9)	94.5 (0.9)	<sup>(c)</sup>

## 4 Conclusions

To solve a classification problem, typically objects are represented by a fixed length feature vector. An alternative is to use a Multiple Instance Learning approach, where objects are represented by a collection (a bag) of feature vectors. This paper shows that for many problems, not just a single feature vector but the full distribution of feature vectors in a bag is informative. Or, in other words, to classify a complicated object like an image, it is often insufficient to look at just a single segment in the image, but you need all segments to get an idea about the image label.

To classify this distribution of vectors, the fitted distribution parameters are used as input features to a standard classifier. Experiments show that a uniform bag distribution performs already well in characterizing these bags. The bag of feature vectors is then represented by the minimum and maximum feature values that appear in the collection of feature vectors. This approach does not only perform well, it is also very cheap and simple to implement and apply. In particular when the number of training bags and the number of instances per bag is limited, this simple representation is sufficiently robust and flexible for a subsequent classification.

More complicated bag distributions can be used as well. Assuming a Gaussian distribution put more heavy demands on the sample size, because a covariance matrix has to be estimated. But using just the mean of the instances (therefore assuming that all Gaussian distribution have the same covariance matrix), gives surprisingly good results and for some problems this approach outperforms all other classifiers. This is most prominent in the (very) high dimensional Web recommendation problem, where most classifiers fail to work well in the 5863 dimensional feature space.

An open issue is still what distribution parameters to use for classification. Not only different distributions can be chosen, also the distributions can be characterized in several equivalent ways (as the equations (6) and (8) for the uniform distribution show). This can result in (slightly) different performances of the final classifier, as is shown in some of the experiments in this paper. Given the relatively low cost of computing this feature representation, it is feasible to use standard crossvalidation to find the optimal representation.

## References

- [1] S. Andrews, T. Hofmann, and I. Tsochantaris. Multiple instance learning with generalized support vector machines. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.
- [2] S. Andrews, I. Tsochantaris, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems 15*, pages 561–568. MIT Press, 2003.
- [3] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. In *VISUAL*, pages 509–516, Amsterdam, 1999.

- [4] Y. Chen, J. Bi, and J.Z. Wang. MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947, 2006.
- [5] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.
- [6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 524–531, 2005.
- [7] T. Gärtner, P.A. Flach, A. Kowwalczyk, and A.J. Smola. Multi-instance kernels. In C. Sammut and A. Hoffmann, editors, *Proceedings of the 19th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 2002.
- [8] S.J. Goldman. SIVAL (spatially independent, variable area, and lighting) benchmark, 1998.
- [9] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International journal of computer vision*, 43(1):29–44, 2001.
- [10] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems*, volume 10, pages 570–576. MIT Press, 1998.
- [11] J.F. Murray, G.F. Forman Hughes, and K. Kreutz-Delgado. Machine learning methods for predicting failures in hard drives: a multiple-instance application. *Journal of Machine Learning Research*, 6:783–816, 2005.
- [12] P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. In *Advances in Neural Inf. Proc. Systems (NIPS 05)*, pages 1419–1426, 2005.
- [13] X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. In *Proc. of the Pacific-Asia conference on knowledge discovery and data mining*. Springer Verlag, 2004.
- [14] Z.-H. Zhou, K. Jiang, and M. Li. Multi-instance learning based web mining. *Applied Intelligence*, 22(2):135–147, 2005.
- [15] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-i.i.d. samples. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 1249–1256, New York, NY, USA, 2009. ACM.

# A Norm Language for Distributed Organizations

Bas Testerink <sup>a</sup>

Mehdi Dastani <sup>b</sup>

<sup>a</sup> *Utrecht University, B.J.G.Testerink@uu.nl*

<sup>b</sup> *Utrecht University, M.M.Dastani@uu.nl*

## Abstract

Existing programming languages for exogenous normative organizations are used to make centralized control structures for multi-agent systems. However, some multi-agent system applications require a distributed control mechanism due to the structure and/or nature of the application. In this paper we address this problem by proposing a programming language for distributed exogenous normative organizations. Our distributed organizations handle norms with a set of sub-organizations which observe and influence a partition of the environment. We will cover the syntax and operational semantics of the proposed programming language.

## 1 Introduction

Agents in a multi-agent system are assumed to be autonomous in the decisions they take. Organizations are used to control the agents. If all the agents would be programmed according to the organizational specification, then we would have an endogenous organization. We however focus on exogenous organizations where the organizational influence is outside of the agents themselves. An exogenous approach to programming organizations supports programming principles such as maintainability, encapsulation and separation of concerns. A specific type of organizations, called exogenous normative organizations (ENO), uses norms to control and coordinate agent activity. Norms describe explicitly what kind of situation would violate the organizational design, and how such violations should be treated. The latter can be done through regimentation and/or enforcement. Regimentation ensures that norm violations are avoided or prevented. Enforcement allows norm violations, but imposes sanctions to compensate for the violation. In a similar vein we could use rewarding for the adherence to norms. We consider norms as conditional obligations and prohibitions with deadlines. The processing of norms in an ENO requires creating/eliminating norms based on their conditions and deadlines, monitoring the activity of agents, evaluating their behavior with respect to the specified norm set and finally determine appropriate consequences for the behavior.

Norms are common within organizational frameworks (cf. Moise[6] and OperA[2]), and we have special languages to program them (for instance the language from [4], 2OPL [1] and NPL/NOPL [5]). The languages can differ in the way the organization is used. In Moise for instance norms are about actions, and sanctions are carried out by agents. As a consequence the framework requires compatible agent technology in order to work (cf.  $\mathcal{J}$ -Moise<sup>+</sup>[7]). In 2OPL norms are about state of the environment, and the organization itself can change the environment as a way of sanctioning. This does not require special compatibility work but makes it harder when we do want to control specific actions. We will use the 2OPL [1] approach in this paper.

Existing programming languages for ENO's generally assume that norms are processed centrally. This is, however, not a realistic assumption for at least three reasons. First, a central norm processing approach is not scalable in the sense that more agents increases the amount of activity to control which in turn decreases the performance of the organization. Second, some applications require a set of ENO's because this would resemble the reality or because different organizations are maintained and modified independently. Third, a centralized organization fails entirely if one error occurs at runtime, which can affect the entire multi-agent system. Because of these reasons, there is a need to extend and develop the existing programming

approaches to facilitate the implementation of distributed ENO's. The distributed nature raises the question if and how different components (i.e., sub-organizations) can interact, and what would be the nature of the interactions.

The need for distributed ENO's can be illustrated by so-called smart roads. These are road systems which are extended with an ICT infrastructure that helps to regulate and manage traffic. We can view such applications as multi-agent systems where the cars are agents (which are black boxes) that utilize the infrastructure. The control and coordination of agents is done through an organization that is based on traffic regulations. In this paper we describe the language to implement that organization. The goal of smart roads is to maximize throughput and road safety. The control and coordination of cars on roads are based on traffic laws and regulations. In this application, the traffic laws (i.e., traffic norms and sanctions) that need to be applied to each road segment may be different (traffic laws for highways are different than for a secondary roads in an urban area) and can be maintained by different people. Also, the norms and sanctions in each part of the road system may change dynamically based on the actual traffic situation in that or other road segments. In smart roads, each road segment is enriched with an organization. The organization exogenously monitors the behavior of cars (using necessary sensors), evaluates them based on the actual norms and regulations, imposes sanctions (i.e., sending a fine to the car owner, changing the maximum speed or redirecting traffic) and, if necessary, modifies actual norms. Obviously, the behavior of all cars in the entire road system cannot be monitored centrally without compromising performance if the system can grow arbitrarily. Note that the above requirements reflect the basic programming principles such as scalability, modularity and encapsulation.

In this paper, we focus on distributed ENO's and propose a programming language that supports their implementation. Our language is not tailored for a specific framework. We had to make some design choices nonetheless. Our language is state based, i.e., it expresses which states ought to be (and not which actions ought to be done). Furthermore, we incorporated temporal aspects such as deadlines and use both rewarding and sanctioning. With our language one can specify sub-organizations that interact with each other and thus form together a distributed ENO.

## 2 Expressiveness of the language

To illustrate the need of various features in our normative language, we consider an example scenario from the smart roads application. Let us assume some highway is partitioned in two road segments A and B, and traffic flows from A to B. To sense the status of the road we have sensors attached to each electronic road sign. In our scenario an accident has occurred at the beginning of segment B. We would like our infrastructure to react to this incident by adjusting the speed regulation for segments A and B. Our example norm in this scenario is that cars ought to keep their velocity lower than the speed which is depicted on the electronic road signs.

In order to control and coordinate the behavior of agents in an open multi-agent system one needs to be able to exert power on agents. In an open multi-agent system, it is impossible to determine the agents' internal decision making mechanism. However, as the designer of a multi-agent platform one can exert power on agents by controlling the entities on which agents depend, i.e., the environment. In this perspective, also presented in [1], agents running on an open platform perform actions (e.g., making requests to use services or resources available on the platform, or to communicate with other agents on that platform) and the organization decides how they are realized. The environment in a smart roads application could be the electronic road signs with sensors, and a registration system for fines. We can store the sensor data in a database. An action a car can perform - intentionally or not - is among others passing a sensor with a certain speed. We want the organization to detect situations like accidents and adjust the speed limit. If an agent passes a sensor with a velocity that is too high, then that agent has to be sanctioned.

A programming language for ENO's must be able to represent and change the environment. We build on the programming language 2OPL [1] and extend it with additional constructs to support the implementation of distributed organizations. The first change is that we partition the environment. The language should be able to represent a partition, which we do with facts. Facts are modified by actions. Action consequences consist sequences of fact assertions and retractions. These sequences are executed in a non-interleaving mode. In our scenario an accident causes adjustment of the speed limitation, which is reflected by the signs on the road. So after an accident the facts should be updated in such a way that the new speed limitation holds and is projected on the road signs.

In order to be able to implement the above issues, we distinguish between norm schemes and norm

instances. Norm schemes can be instantiated when their precondition is satisfied. An instantiated norm creates a deontic influence. We limit ourselves in this paper to obligations and prohibitions. We do not bind our deontic operators to actions, but to the state of the environment. If something which is obliged is not brought about, or if something which is forbidden is brought about, then this counts as a violation of the norm. Otherwise it counts as an adherence to the norm. For the violation of obligations and the adherence to prohibitions we need deadlines. We can also make use of expiration clauses. The difference between deadlines and expiration clauses is that a deadline is used to generate either an adherence or violation effect, while an expiration removes the norm instance without any consequences.

We mentioned as an example norm that cars ought to drive slower than the speed indication on the road signs. After an accident the speed indication changes. We can use the passing of a road sign, and the fact that the sign displays an adjusted limit, as a precondition which instantiates this norm for an individual car. The deontic influence is that the car is obliged to have the lower velocity. To see if a car is in violation we need a deadline. Cars are notified of the speed limit when they pass a road sign. So when the speed limit is adapted after the accident, the cars ought to have the adjusted velocity at the road sign after the next. An expiration clause would be that the sensor system fails or that the accident site is cleared. A violation effect for our example could be a fine.

A novel feature of our extension is the use of labeled literals in norms and action specifications. Because sub-organizations only partially observe the environment (i.e., each organization operates on one partition), they might need to query and influence other organizations as well. We require that each organization has a unique label that can be used to query a fact in its environment partition or to update it. The programmer only has to type the label and the interpreter then handles all the necessary interactions to get the right information. In our scenario segment B, where the accident occurred, changes both its own road signs and those of segment A. Also, if a car is notified of the new speed at the last sensor of segment A, then it should have adapted its speed at the first sensor of segment B.

### 3 Syntax

A distributed ENO can be implemented by programming a set of separate organizations. An organization can be implemented by programming the initial state of the environment partition on which it operates, the set of norms that can be enforced by the organization, and the set of action specifications that describe the effects of agents' actions. We view norms as consisting of several (optional) attributes. The parts of a norm are: a name, a precondition, a prohibited state or an obligated state, a deadline, an expiration clause, a consequence for violation and a consequence for obeying the norm. Any omitted attributes of a norm are set to `false`, with the exception of the precondition which is set to `true`. The proposed syntax in this paper states an attribute after which its value is given. We can now keep using the comma for conjunction - as in Prolog - and just leave out an attribute if we want to give it a standard value. Any literals concerning other organizations are notated with a label. Literals without labels refer to the organization itself. Those with labels refer to organizations that are identified by the label. With the positive literal  $\$a:p$  we indicate the positive literal  $p$  from the model of organization  $a$ . With the fact assertion  $\$a:+p$  we indicate that organization  $a$  should add  $p$  to its model. The EBNF syntax specification is given in Figure 1.

Example constructs are displayed in Figure 2. There we see a possible organization that is assigned to a highway segment. The norm is a simplified speed limit norm. We assume that the organization it applies to has a downstream neighboring segment that has an organization which is identified by the label `segment23`. What happens is that if a car passed the last sensor of this segment, then it should drive according to the speed limit at the first sensor of `segment23`. In a real application one would make this norm more general to capture the regulation that between two sensors a car should adapt its speed, but we keep the example concise. The fine for the violation is effectuated in another organization that represents a fine registration system, called `fineDB`. The norm expires if the sensors of the next segment fail (if those of the current segment fail, then we would not even notice that the last sensor was passed). The intuitive reading of the conjunction  $\$segment23:passedFirst(C), exceeds(C, L)$  is 'in `segment23` it holds that  $C$  has passed the first sensor, and  $C$ 's velocity exceeds  $L$ '. The meaning of  $\$fineDB:+fine(C, speeding, 100)$  is 'it now holds in `fineDB` that  $C$  has a fine for speeding of 100 euro's'.

<code>&lt;ORG&gt;</code>	<code>::= "Brute facts:" &lt;ATOM&gt;* "Actions:" &lt;ACTIONSPEC&gt;* "Norms:" &lt;NORM&gt;*</code>
<code>&lt;ACTIONSPEC&gt;</code>	<code>::= "action{" &lt;HEAD&gt; [ &lt;PRECONDITION&gt; ] &lt;POSTCONDITION&gt; "}"</code>
<code>&lt;HEAD&gt;</code>	<code>::= "head:" &lt;ATOM&gt; ""</code>
<code>&lt;POSTCONDITION&gt;</code>	<code>::= "postcondition:" &lt;MOD&gt; (";" &lt;MOD&gt;)* ""</code>
<code>&lt;MOD&gt;</code>	<code>::= [ &lt;LABEL&gt; ] ("+" "-" &lt;ATOM&gt;)</code>
<code>&lt;QUERY&gt;</code>	<code>::= &lt;LITERAL&gt; (";" &lt;LITERAL&gt;)* ""</code>
<code>&lt;LITERAL&gt;</code>	<code>::= [ &lt;LABEL&gt; ] [ "not" ] &lt;ATOM&gt;</code>
<code>&lt;LABEL&gt;</code>	<code>::= "\$" ( &lt;ATOM&gt;   &lt;VAR&gt; ) ""</code>
<code>&lt;NORM&gt;</code>	<code>::= "norm{" &lt;NAME&gt; [ &lt;PRECONDITION&gt; ] [ ( &lt;PROHIBITION&gt; )   ( &lt;OBLIGATION&gt; ) ] [ ( &lt;DEADLINE&gt; ) ] [ ( &lt;EXPIRATION&gt; ) ] [ ( &lt;VIOLATED&gt; ) ] [ ( &lt;OBEYED&gt; ) ] "}"</code>
<code>&lt;NAME&gt;</code>	<code>::= "name:" &lt;ATOM&gt; ""</code>
<code>&lt;PRECONDITION&gt;</code>	<code>::= "precondition:" &lt;QUERY&gt;</code>
<code>&lt;PROHIBITION&gt;</code>	<code>::= "prohibition:" &lt;QUERY&gt;</code>
<code>&lt;OBLIGATION&gt;</code>	<code>::= "obligation:" &lt;QUERY&gt;</code>
<code>&lt;DEADLINE&gt;</code>	<code>::= "deadline:" &lt;QUERY&gt;</code>
<code>&lt;EXPIRATION&gt;</code>	<code>::= "expiration:" &lt;QUERY&gt;</code>
<code>&lt;VIOLATED&gt;</code>	<code>::= "violated:" &lt;MOD&gt; (";" &lt;MOD&gt;)* ""</code>
<code>&lt;OBEYED&gt;</code>	<code>::= "obeyed:" &lt;MOD&gt; (";" &lt;MOD&gt;)* ""</code>

Figure 1: Proposed syntax for writing norms. Atoms are first-order atoms and may contain variables. Variables are notated as Prolog variables (starting with an upper case character or underscore).

<pre>Brute facts limit(120). passedLast(car1). broken(car3).</pre>	<pre>Actions: action {   head: accident.   pre: limit(L).   post: -limit(L);         +limit(50). }</pre>	<pre>Norms: norm {   name: cross_segment_speed_limit.   pre: passedLast(C), limit(L).   prohib: \$segment23:passedFirst(C), exceeds(C,L).   deadln: accident_cleared.   exp: \$segment23:broken(sensors).   viol: \$fineDB:+fine(C, speeding, 100).</pre>
--	--	---

Figure 2: Some facts, an action, and a norm scheme. Attributes are abbreviated.

## 4 Operational semantics

Operational semantics provide meaning to the syntax of a programming language. We use labeled transition systems from [10]. Transitions are descriptions of the way in which the state of the system is changed by performing an operation of the programming language. We discriminate between organization transitions, and distributed organization transitions. But first we start with some necessary definitions and functions.

### 4.1 Preliminary definitions

When we program norms, we actually program norm schemes, i.e., abstract norms that need to be instantiated to create deontic influence. All attributes of norms are stored in tuples. To keep semantic rules short we use the notation  $ns_{att}$  to indicate the value of the attribute  $att$  from the norm scheme  $ns$ . A norm scheme  $ns$  is uniquely instantiated by using  $ns_{name}$  and the substitution for  $ns_{precondition}$ . This substitution should instantiate all variables in a norm scheme. To formalize this we define besides norm schemes also their well-formedness.

**Definition 1. Norm scheme** A norm scheme  $ns$  is a tuple  $\langle name, precondition, prohibition, obligation, deadline, expiration, violated, obeyed \rangle$ .  $ns_{name}$  is an atom.  $ns_{precondition}$ ,  $ns_{prohibition}$ ,  $ns_{obligation}$ ,  $ns_{deadline}$  and  $ns_{expiration}$  are conjunctions of literals.  $ns_{violated}$  and  $ns_{obeyed}$  are sequences of fact assertions and retractions.

A well-formed norm scheme should satisfy two constraints. First, either the prohibition or the obligation (not both) formula should be  $\perp$ . Second, all variables should be instantiated by the substitution resulted from the precondition.

**Definition 2. Well-formedness of norm schemes** Given a norm scheme  $ns = \langle name, precondition, prohibition, obligation, deadline, expiration, violated, obeyed \rangle$ , the set of variables  $\overline{v_1}$  that occur in  $ns_{precondition}$ , and the set of variables  $\overline{v_2}$  that occur in  $ns_{precondition}$ ,  $ns_{prohibition}$ ,  $ns_{obligation}$ ,  $ns_{deadline}$  and  $ns_{expiration}$ ,  $ns$  is well-formed iff  $\overline{v_2} \subseteq \overline{v_1}$  and either  $ns_{prohibition} = \perp$  or  $ns_{obligation} = \perp$ , but not both.

The configuration (state) of an organization is represented by a tuple consisting of a set of facts representing its environment partition, a set of action specifications representing the effects of agents' actions

on its environment partition, a set of norms schemes, a set of instances of norm schemes, and the actions that are performed by the agents in the organization. Facts are first order literals. Action specifications are triples consisting of a head, a precondition (conjunction of literals) and a postcondition (sequence of assertions/retractions). A norm instance is a tuple containing a norm scheme and the substitution which made the precondition of the scheme entailed by the facts at the moment of instantiation. Actions are assumed to be stored in a queue.

**Definition 3. Organization configuration** *The configuration of an organization is a tuple  $\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle$ , where  $\iota$  is a unique identifier,  $\Sigma$  is a set of action specifications,  $\Delta$  is a set of well formed norm schemes,  $\delta$  is a set of norm instantiations,  $\sigma$  is a set of ground positive first order literals representing the environment partition, and  $\xi$  is a queue of ground positive first order literals, which represent the actions. The initial configuration of an organization is a tuple  $\langle \iota, \Sigma, \Delta, \emptyset, \sigma, [] \rangle$ .*

A distributed organization is in essence a set of organizations. Other natural properties would be roles, power, responsibility, delegation structure and so forth. But for our purposes we consider only the sub-organizations.

**Definition 4. Distributed organization** *The configuration (state) of a distributed organization is  $\mathbb{O} = \{O_1, \dots, O_n\}$ , where  $O_i$  is the configuration of an organization.*

We need an entailment operator for deriving whether a certain formula is entailed by the configuration of an organization. Labels used in the formula indicate literals that are stored elsewhere. Our entailment operator is notated as  $O \models \varphi\theta$ , indicating that the organization  $O = \langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle$  entails  $\varphi$  under substitution  $\theta$ . A label can be an atom with variables, or a variable itself. These variables are handled equally as other variables. Thus  $\ell\theta$  indicates the label under substitution theta. The definition of  $\models$  is shown in Table 1.

$\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models \varphi\theta$	$\Leftrightarrow^1$	$\varphi\theta \in \sigma$ ( $^1$ only if $\varphi \neq \$\ell:\psi$ )
$\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models \text{not } \varphi$	$\Leftrightarrow^2$	$\nexists \theta : \varphi\theta \in \sigma$ ( $^2$ only if $\varphi \neq \$\ell:\psi$ )
$\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models (\$ \ell:\varphi)\theta$	$\Leftrightarrow$	$\langle \ell\theta, \Sigma', \Delta', \delta', \sigma', \xi' \rangle \models \varphi\theta$
$\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models \text{not } \$\ell:\varphi$	$\Leftrightarrow$	$\nexists \theta: \langle \ell\theta, \Sigma', \Delta', \delta', \sigma', \xi' \rangle \models \varphi\theta$
$\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models (\varphi(\bar{x}) \wedge \psi(\bar{y}))\theta$	$\Leftrightarrow$	$\exists \theta_1 : [\theta_1 = \theta \bar{x} \text{ and } \langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models \varphi\theta_1 \text{ and } \exists \theta_2 : [\theta_2 = \theta (\bar{y} \setminus \bar{x}) \text{ and } \langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models \psi\theta_1\theta_2]]$
$\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models (\varphi \vee \psi)\theta$	$\Leftrightarrow$	$\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models \varphi\theta \text{ or } \langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models \psi\theta$

Table 1: Definition of the entailment operator. ‘|’ is read as ‘restricted to the domain’. ‘ $\varphi(\bar{x})$ ’ is read as ‘formula  $\varphi$  which variables form the set  $\bar{x}$ ’.

In distributed organizations, updates in one organization may require updates in other organizations as well. In order to apply a sequence of updates, we define a function *update* that given a fact base  $\sigma$  and a sequence of modifications  $\Pi$  returns a new fact base. Each modification either removes or adds a fact. A fact modification  $\pi$  can be labeled in which case it is represented as  $\$ \ell : \phi$ , where  $\phi$  is either  $+\rho$  or  $-\rho$ , and  $\rho$  is a fact. A sequence of modifications  $\Pi$  is represented as  $[\pi_0; [\dots; [\pi_n; []] \dots]]$ . Let  $\Pi^*$  denote the set of possible modification sequences. The update function *update* :  $\sigma^* \times \Pi^* \rightarrow \sigma^*$  is defined as follows.

$$\text{update}(\sigma, \Pi) = \begin{cases} \text{update}(\sigma \cup \{\rho\}, \Pi') & \Pi = [+ \rho; \Pi'] \ \& \ \rho \neq \$ \ell : \psi \\ \text{update}(\sigma \setminus \{\rho\}, \Pi') & \Pi = [- \rho; \Pi'] \ \& \ \rho \neq \$ \ell : \psi \\ \text{update}(\sigma, \Pi') & \Pi = [ \$ \ell : + \rho; \Pi'] \\ \text{update}(\sigma, \Pi') & \Pi = [ \$ \ell : - \rho; \Pi'] \\ \sigma & \Pi = [] \end{cases}$$

Note that in *update* only non-labeled modifications are taken into account. If a sequence is received from another organization, then we need to extract the relevant modifications from it. The function *extract* does this. Given a label  $\ell$  and a sequence  $\Pi$ , *extract* returns the unlabeled sequence of modifications in  $\Pi$  with label  $\ell$ . Let  $L$  denote the set of possible labels. The extract function *extract* :  $L \times \Pi^* \rightarrow \Pi^*$  is defined as

follows.

$$\text{extract}(\ell, \Pi) = \begin{cases} [+ \rho; \text{extract}(\$ \ell, \Pi')] & \Pi = [\$ \ell : + \rho; \Pi'] \\ [- \rho; \text{extract}(\$ \ell, \Pi')] & \Pi = [\$ \ell : - \rho; \Pi'] \\ \text{extract}(\$ \ell, \Pi') & \Pi = [+ \rho; \Pi'] \\ \text{extract}(\$ \ell, \Pi') & \Pi = [- \rho; \Pi'] \\ \text{extract}(\$ \ell, \Pi') & \Pi = [\$ \ell' : + \rho; \Pi'] \ \& \ \ell' \neq \ell \\ \text{extract}(\$ \ell, \Pi') & \Pi = [\$ \ell' : - \rho; \Pi'] \ \& \ \ell' \neq \ell \\ \square & \Pi = \square \end{cases}$$

Norm instances can be cleared from the configuration of an organization because their deontic content is satisfied/violated when their deadlines arrive or because they are expired. Two tasks must be performed to clear a norm: first check whether the deontic content, deadline or expiration holds and then modify the configuration appropriately. The *can\_clear* function returns, given an organization configuration and a norm instantiation, whether the instantiation can be cleared. Let  $O^*$  denote the set of all possible organization configurations and  $\langle ns, \theta \rangle^*$  denote the set of possible norm schemes in combination with their possible substitutions. The function  $\text{can\_clear} : O^* \times \langle ns, \theta \rangle^* \rightarrow \{true, false\}$  can be defined as follows.

$$\text{can\_clear}(O, \langle ns, \theta \rangle) = \begin{cases} true & O \models (ns_{prohibition} \vee ns_{obligation} \vee \\ & ns_{deadline} \vee ns_{expiration})\theta \\ false & otherwise \end{cases}$$

Moreover, given an organization configuration  $O$  and a norm instantiation, the function *mod* returns the appropriate modification sequence. If the norm is expired, then the sequence is empty. Otherwise it is checked whether the norm was obeyed or violated. The function  $\text{mod} : O^* \times \langle ns, \theta \rangle^* \rightarrow \Pi^*$  is defined as follows.

$$\text{mod}(O, \langle ns, \theta \rangle) = \begin{cases} \square & O \models ns_{expiration}\theta \\ ns_{obeyed}\theta & O \not\models ns_{expiration}\theta \ \& \ O \models ns_{obligation}\theta \\ ns_{obeyed}\theta & O \not\models ns_{expiration}\theta \ \& \ ns_{prohibition} \neq \perp \ \& \ O \not\models ns_{prohibition}\theta \\ ns_{violated}\theta & O \not\models ns_{expiration}\theta \ \& \ ns_{obligation} \neq \perp \ \& \ O \not\models ns_{obligation}\theta \\ ns_{violated}\theta & O \not\models ns_{expiration}\theta \ \& \ O \models ns_{prohibition}\theta \end{cases}$$

## 4.2 Transition Rules

What follows are the organization transitions when norms and actions are handled.

If an organization  $O$  receives an update sequence  $\Psi$ , then the proper sequence is extracted from  $\Psi$  and applied to the local fact base of  $O$ . The result is configuration  $O'$ . This transition is denoted by  $O \xrightarrow{\Psi?}_{org} O'$ , where  $\Psi?$  is used to indicate that the transition takes place by receiving  $\Psi$ .

$$\frac{\sigma' = \text{update}(\sigma, \text{extract}(\iota, \Psi))}{\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \xrightarrow{\Psi?}_{org} \langle \iota, \Sigma, \Delta, \delta, \sigma', \xi \rangle} \quad (\text{update facts})$$

A norm scheme of an organization can be instantiated when the configuration of the organization entails its precondition. The norm instance is then added to the set of norm instances.

$$\frac{\delta' = \delta \cup \{ \langle ns, \theta \rangle \mid ns \in \Delta \ \& \ \langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \models ns_{precondition}\theta \}}{\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \rightarrow_{org} \langle \iota, \Sigma, \Delta, \delta', \sigma, \xi \rangle} \quad (\text{instantiate norms})$$

The following transition rule is to clear norm instances. For a norm instance  $ni$  we first check whether  $ni$  can be cleared. Then we determine the consequences which is a sequence of fact modifications. The sequence is applied to the fact base and is also broad casted. The broadcast is denoted by adding  $\Pi!$  to the transition. All the other organizations receive and extract the subsequences for their fact bases and make a transition. This is guaranteed by the *modification synchronization* transition rule presented later on. Note that those organizations which have no labeled modifications for themselves in the sequence can still make a transition. Their subsequence from *update* will be empty and does not change their fact base. Finally the norm instance can be removed. We can clear all clear-able norm instances by repeating this operation until no transition can occur. Let  $O = \langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle$ , the following transition will clear norm instances.

$$\frac{ni \in \delta \ \& \ \text{can\_clear}(O, ni) \ \& \ \Pi = \text{mod}(O, ni) \ \& \ \sigma' = \text{update}(\sigma, \Pi) \ \& \ \delta' = \delta \setminus \{ni\}}{\langle \iota, \Sigma, \Delta, \delta, \sigma, \xi \rangle \xrightarrow{\Pi!}_{org} \langle \iota, \Sigma, \Delta, \delta', \sigma', \xi \rangle} \quad (\text{clear norm})$$

Actions are added to the queue  $\xi$ . We assume that for each action an applicable action specification is present. Future research might include exceptions. We reuse the earlier mentioned *update* function. In the following rules  $\epsilon$  is used to indicate an action.  $\epsilon$  is a ground positive first order literal.

$$\frac{\langle \varphi, \alpha, \psi \rangle \in \Sigma \ \& \ \epsilon = \alpha\theta \ \& \ \langle \iota, \Sigma, \Delta, \delta, \sigma, \epsilon : \xi \rangle \models \varphi\theta\tau \ \& \ \sigma' = \text{update}(\sigma, \psi\theta\tau)}{\langle \iota, \Sigma, \Delta, \delta, \sigma, \epsilon : \xi \rangle \xrightarrow{\psi\theta\tau^1}_{org} \langle \iota, \Sigma, \Delta, \delta, \sigma', \xi \rangle} \quad (\text{perform action})$$

#### 4.2.1 Distributed organization transitions

The distributed organization as a whole changes when its sub-organizations change. On this level we can also synchronize transitions. The first transition describes how the distributed organization changes if an sub-organization makes an internal transition, such as instantiating norms.

$$\frac{O \in \mathbb{O} \ \& \ O \rightarrow_{org} O' \ \& \ \mathbb{O}' = (\mathbb{O} \setminus \{O\}) \cup \{O'\}}{\mathbb{O} \rightarrow_{d-org} \mathbb{O}'} \quad (\text{sub-organization operation})$$

We use  $O \xrightarrow{\Psi^1}_{org} O'$  to notate that organization  $O$  broadcasts a sequence of fact modifications  $\Psi$ . If this transition occurs, then the receiving organizations have to handle this sequence and make a transition, which is notated as  $O \xrightarrow{\Psi^?}_{org} O'$ .

$$\frac{O_i \in \mathbb{O} \ \& \ O_i \xrightarrow{\Psi^1}_{org} O'_i \ \& \ \forall O_j \in \mathbb{O} \setminus \{O_i\} : O_j \xrightarrow{\Psi^?}_{org} O'_j}{\{O_0, \dots, O_{i-1}, O_i, O_{i+1}, \dots, O_k\} \rightarrow_{d-org} \{O'_0, \dots, O'_{i-1}, O_i, O'_{i+1}, \dots, O'_k\}} \quad (\text{modification synchronization})$$

## 5 Related work

In related work there are various methods presented to partition the environment. In [9] the environment, which consists of artifacts, is partitioned in work spaces. Regulations take the form of counts-as and enact rules. Their norms cannot span multiple work spaces. This approach is similar to [3] where norms are handled in a distributed manner by dividing them among normative scenes. Each scene is related to an interaction protocol for an activity. Communication between scenes is made possible by normative transition rules. Our approach is state oriented and lets the programmer decide how to group norms together in sub-organizations. Another distributed approach is presented in [8]. There the agents are assigned to controllers. Controllers contain a law and act as mediators. Their system is also event/action based. A big difference with other approaches is that they divide the agent population rather than the environment.

Related work on normative languages can be found among others in [4], [1] and [5]. In [4] there is also a norm language plus its interpreter presented. Their language consists solely of implication rules. Such a rule has on the left hand side a formula about the model, and the right hand side consists of a conjunction of modifications. 2OPL (organization oriented programming language) [1] uses similar implication rules. In 2OPL there are two kinds of norm related constructs: counts-as rules and sanction rules. Counts-as rules are used to determine what system states constitute what kind of violations. With the sanction rules these violations are coupled to model changes. In [11] temporal norms consisting of a precondition, a deontic influence and a deadline were introduced to replace 2OPL's counts-as rules. Another norm language is NPL (Normative Programming Language)[5]. NPL norms are labeled counts-as rules. Instead of modification on the right hand side, NPL norms only create obligations or fail. Thus if we want something in the environment changed as a consequence of a norm, then a submissive and able agent has to become obliged to make the change. This view is very different from the one in this paper where we consider the normative process to be able to act.

## 6 Conclusions & Future work

Current normative languages consider mainly centralized execution of norms. This view might not be feasible in terms of scalability, but can also ill fit the nature of one's organization. A solution to this problem is to create a network of sub-organizations, and thus construct a distributed organization. In this paper we

discussed a normative language which can be used for implementing distributed organizations. The language can be used stand alone, meaning that the programmer does not have to rely on design tools. With the language one can program norms in a declarative manner and with temporal features. Norms are programmed by using system states rather than actions. A special feature of the language is the possibility of organizations to query and modify each other's environment partition. This is ideal if the organization has a clear modular structure, but in some cases has norms that overlap modules.

We still need to analyze the full theoretic consequences of the language. A distributed organization has different properties than a centralized one due to the parallel execution of the sub-organizations. We also want to provide guidelines on how to use the language. Currently one might question the idea that any organization can query and change anything in another organization. Also maintenance will be harder if you have to copy the same norm in multiple organizations. For these latter problems we want to develop an IDE to help during development and maintenance. Other directions include describing the inter organizational communication and exploring different modes of distribution. We now choose geographic distribution, but we can also use distribution based on functionality or the organization's topology.

## References

- [1] M. Dastani, D. Grossi, J-J. Ch. Meyer, and N. Tinnemeier. Normative multi-agent programs and their logics. In KRAMAS, pages 16–31, 2008.
- [2] V. Dignum. A model for organizational interaction: based on agents, founded in logic. PhD thesis, Universiteit Utrecht, 2004.
- [3] Dorian Gaertner, Andres Garcia-Camino, Pablo Noriega, J.-A. Rodríguez-Aguilar, and Wamberto Vasconcelos. Distributed norm management in regulated multiagent systems. In Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, AAMAS '07, pages 90:1–90:8, New York, NY, USA, 2007. ACM.
- [4] A. García-Camino, J. Rodríguez-Aguilar, C. Sierra, and W. Vasconcelos. Constraint rule-based programming of norms for electronic institutions. Autonomous Agents and Multi-Agent Systems, 18:186–217, 2009.
- [5] J. Hübner, O. Boissier, and R. Bordini. A normative programming language for multi-agent organisations. Annals of Mathematics and Artificial Intelligence, 62:27–53, 2011.
- [6] J. Hübner, J. Sichman, and O. Boissier.  $S - \text{Moise}^+$ : A middleware for developing organised multi-agent systems. In O. Boissier, J. Padget, V. Dignum, G. Lindemann, E. Matson, S. Ossowski, J. Sichman, and J. Vázquez-Salceda, editors, Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems, volume 3913 of Lecture Notes in Computer Science, pages 64–77. Springer Berlin / Heidelberg, 2006.
- [7] Jomi F. Hübner, Jaime S. Sichman, and Olivier Boissier. Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels. Int. J. Agent-Oriented Softw. Eng., 1(3/4):370–395, December 2007.
- [8] Naftaly H. Minsky and Victoria Ungureanu. Law-governed interaction: a coordination and control mechanism for heterogeneous distributed systems. ACM Trans. Softw. Eng. Methodol., 9(3):273–305, July 2000.
- [9] Michele Piunti, Olivier Boissier, Jomi Fred Hübner, and Alessandro Ricci. Embodied organizations: a unifying perspective in programming agents, organizations and environments. In Olivier Boissier, Amal El Fallah-Seghrouchni, Salima Hassas, and Nicolas Maudet, editors, MALLOW, volume 627 of CEUR Workshop Proceedings. CEUR-WS.org, 2010.
- [10] G. D. Plotkin. A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19, University of Aarhus, 1981.
- [11] N. Tinnemeier. Organizing agent organizations : syntax and operational semantics of an organization-oriented programming language. PhD thesis, Utrecht University, SIKS, 2011.

# Intercausal Cancellation in Bayesian Networks

Steven P.D. Woudenberg<sup>a</sup>    Linda C. van der Gaag<sup>a</sup>    Carin M.A. Rademaker<sup>b</sup>

<sup>a</sup> *Department of Information and Computing Sciences, Utrecht University*

<sup>b</sup> *Department of Clinical Pharmacy, University Medical Center, Utrecht University*

## Abstract

When constructing Bayesian networks with domain experts, knowledge engineers often use the noisy-OR model to alleviate the burden of probability elicitation. In a recent study, we have shown that ill-considered use of this causal interaction model can substantially decrease a network's performance, especially for domains in which causal mechanisms include mutual cancellation effects. Motivated by this observation, we designed a parameterised interaction model to capture different types of intercausal cancellation. We detail properties of our model and illustrate its use in the real-world domain of pharmacology.

## 1 Introduction

Bayesian networks of realistic size easily require hundreds and sometimes even thousands of probabilities for their conditional probability tables. While for some application domains these probabilities are readily available or can be estimated from data, for other domains they need to be assessed by experts. This assessment task is quite time consuming, and is often impeded by the experts feeling uncomfortable with providing numbers to describe their knowledge and experience. Over the last decades, researchers have proposed different approaches to alleviating the burden of probability elicitation, by creating tailored elicitation methods [1] and by designing causal interaction models for general use [2, 3, 4].

A causal interaction model in essence is a parameterised conditional probability table for the effect variable of a causal mechanism in a Bayesian network. The best-known interaction model is the noisy-OR model, originally designed by J. Pearl [5]. This model requires explicit assessment of just a small number of conditional probabilities, for each modelled cause separately; these probabilities constitute the model's parameters. The other probabilities required to arrive at a fully specified conditional probability table need not be assessed explicitly, but are computed from the model's parameters through simple mathematical functions; these functions assume a specific pattern of interaction among the causes of the common effect in the mechanism under study. Use of the noisy-OR model incurs a substantial reduction of the number of probabilities to be assessed by experts, from exponential to linear in the number of causes of an effect; in addition, the probabilities which are typically hardest to assess are now calculated by the model.

Over the years, evidence has built up that the noisy-OR model can be used in many real-world applications of Bayesian networks, even if its underlying assumptions are not met [6, 7]. In a recent study however, we demonstrated that ill-considered use of the model can result in poorly calibrated probability values [8]. Especially for causal mechanisms involving cancellation effects among their cause variables, will the conditional probabilities computed by the noisy-OR model deviate substantially from the true probabilities. We encountered such cancellation effects upon describing interactions among pharmaceutical substances. Using noisy-OR computed probabilities in this domain of pharmacology would result in these cancellation effects not being properly modelled and possibly in erroneous conclusions being drawn.

In this paper, we study causal mechanisms embedding cancellation effects among their cause variables. We distinguish between different types of cancellation effect, ranging from full to partial cancellation and from one-sided to mutual cancellation. Motivated by the examples encountered in the real-world domain of pharmacology, we design a new causal interaction model and show that it embeds the various types of cancellation effect discerned. This cancellation model is designed from first principles and can be considered a parameterised conditional probability table, just like the noisy-OR model. The cancellation model requires

expert assessments for essentially the same conditional probabilities as the noisy-OR model. In addition however, it requires a small number of tuning parameters, for which we suggest a default value for general use. We demonstrate how our cancellation model is applied for two real-world examples in pharmacology.

The paper is organised as follows. In Section 2, we introduce our notational conventions and review the noisy-OR model; in addition, we describe the qualitative concepts of probability used throughout the paper. We elaborate on our motivation for defining an intercausal cancellation model in Section 3. Section 4 defines our model and describes its underlying assumptions and properties. In Section 5, we demonstrate application of the model for our motivating examples. The paper is concluded with Section 6.

## 2 Preliminaries

We consider a joint probability distribution  $\Pr$  over a set of random variables. Each variable  $V$  is assumed to be binary, with values  $\bar{v}$  and  $v$  describing the absence or presence of some concept respectively. We assume that the distribution  $\Pr$  is modelled by a Bayesian network. Within this network, we consider a causal mechanism. For ease of notation, we assume that the mechanism consists of two cause variables  $A$  and  $B$ , and a single effect variable  $E$ , as shown in Figure 1 on the left. The strengths of the probabilistic influences of the two cause variables on the effect variable are expressed by means of a conditional probability table, which describes the probability distributions  $\Pr(E \mid A, B)$  over  $E$  for each possible combination of values of the cause variables  $A$  and  $B$ ; the table thus specifies four distributions.

The noisy-OR model now provides a parameterised conditional probability table for the effect variable  $E$  of the causal mechanism. The model captures the information that each modelled cause in essence suffices to give rise to the effect  $e$ , but that some unmodelled processes may inhibit the effect to occur; it essentially models a logical OR with uncertain perturbation effects. The parameters of the model are the probabilities

$$\begin{aligned} x &= \Pr(e \mid \bar{a}, b) \\ y &= \Pr(e \mid a, \bar{b}) \end{aligned}$$

of the effect  $e$  arising in the presence of just a single cause. These probabilities express that in the presence of the cause  $a$  for example, the effect  $e$  is inhibited with probability  $1 - y$  by some unmodelled process.  $\Pr(e \mid \bar{a}, \bar{b})$  is taken to be zero by the model. To arrive at a fully specified probability table for the variable  $E$ , the conditional probability of the effect  $e$  occurring given the presence of both causes is taken to be

$$\Pr(e \mid a, b) = 1 - (1 - x) \cdot (1 - y) = x + y - x \cdot y$$

Underlying the noisy-OR model are the properties of accountability and exception independence. The property of accountability states that the effect  $e$  cannot occur as long as its causes are absent, that is  $\Pr(e \mid \bar{a}, \bar{b}) = 0$ . The property of exception independence pertains to the exception mechanisms that may inhibit the effect to arise in the presence of a cause. The arc  $A \rightarrow E$  can be viewed as an essentially deterministic causal relation which has associated an inhibitor variable  $I_a$  to describe the uncertainty involved; a similar observation holds for the arc  $B \rightarrow E$ . Exception independence now states that the inhibitor variables  $I_a$  and  $I_b$  of the two causes are mutually independent. The conceptual model thus underlying the noisy-OR model is shown in Figure 1 on the right: the inhibitor variables have now been made explicit, and the variables  $X_a$  and  $X_b$  capture the processes through which the causes  $a$  and  $b$  achieve the effect  $e$ .

In the sequel, we use concepts of qualitative probability to study properties of causal interaction models. We briefly review the concepts of qualitative influence and of additive and product synergy, stating them in terms of our mechanism under study. A qualitative influence between two variables in general expresses how the value of the one variable influences the probabilities of the values of the other variable. A positive qualitative influence of the cause variable  $A$  on the effect variable  $E$  for example, denoted  $S^+(A, E)$ , ex-

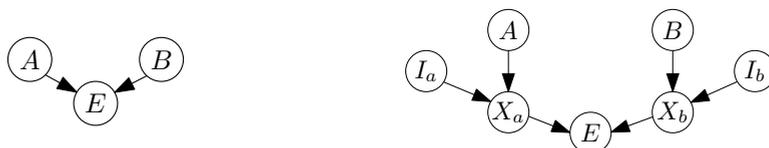


Figure 1: A causal mechanism with the cause variables  $A$  and  $B$ , and the effect variable  $E$  (left). The conceptual model underlying the noisy-OR model, with the inhibitor variables made explicit (right).

presses that observing the value  $a$  for  $A$  makes the value  $e$  for  $E$  more likely, regardless of the influence of  $B$  on  $E$ , that is,

$$\Pr(e \mid a, B) - \Pr(e \mid \bar{a}, B) \geq 0$$

for any value of  $B$ . A negative influence  $S^-$  and a zero influence  $S^0$  are defined analogously. The influence of  $A$  on  $E$  is non-monotonic, denoted by  $S^?$ , if its sign is dependent upon the value of  $B$ . The importance of the qualitative influence lies in the property that evidence for the variable  $A$  will upon inference cause a shift in the probability distribution of  $E$ , in the direction indicated by its associated sign.

An additive synergy expresses how two variables in general interact in their joint influence on a third variable. A positive additive synergy of the two cause variables  $A$  and  $B$  on the effect variable  $E$ , denoted  $Y^+(\{A, B\}, E)$ , expresses that the joint influence of  $A$  and  $B$  on  $E$  exceeds the sum of their separate influences, that is,

$$\Pr(e \mid a, b) + \Pr(e \mid \bar{a}, \bar{b}) \geq \Pr(e \mid a, \bar{b}) + \Pr(e \mid \bar{a}, b)$$

Negative and zero additive synergies are defined analogously; in our causal mechanism with two cause variables only, the additive synergy cannot be non-monotonic. Product synergies in general express how the value of one cause variable influences the probability distribution of another cause variable given a value for their common effect variable. A positive product synergy of  $A$  on  $B$  (and vice versa) given the value  $e$  for the common effect variable  $E$  for example, denoted  $X^+(\{A, B\}, e)$ , expresses that, given  $e$ , the value  $a$  for  $A$  renders the value  $b$  for  $B$  more likely, which amounts to

$$\Pr(e \mid a, b) \cdot \Pr(e \mid \bar{a}, \bar{b}) \geq \Pr(e \mid a\bar{b}) \cdot \Pr(e \mid \bar{a}b)$$

Negative and zero product synergies are defined analogously; for causal mechanisms involving two cause variables only, the product synergy cannot be non-monotonic. The importance of the product synergy lies in the property that, upon observing a specific value for the common effect variable, a qualitative intercausal influence is induced between the cause variables  $A$  and  $B$ , with the synergy's sign.

### 3 Motivation

The noisy-OR model has met with wide-spread use because of its evident advantages for the quantification task. Various empirical studies have shown that networks with noisy-OR calculated probability tables perform comparably to expert-quantified ones [2, 6, 7], even when domain knowledge does not match the patterns of intercausal interaction imposed by the model. The consistent conclusions from these studies have led to the suggestion that Bayesian networks are quite robust against the changes induced in their conditional probability tables by using the noisy-OR model, and appear to warrant the model's use for mere pragmatic reasons also to causal mechanisms which do not exhibit the assumed interactions.

In our recent work, we investigated the effects of using the noisy-OR model in further detail, for causal mechanisms capturing intercausal interactions that differ from the model's assumptions [8]. More specifically, we used sensitivity analysis techniques to study the effects of changing the noisy-OR computed probabilities. As an example, the graph shown in Figure 2 on the left describes the output probability  $\Pr(a \mid b, e)$  in terms of the input probability  $\Pr(e \mid a, b)$ , given specific values for  $\Pr(e \mid a, \bar{b})$  and  $\Pr(e \mid \bar{a}, b)$ ; the left-most dot in the graph represents the noisy-OR calculated value for the input probability and the rightmost dot marks its true value. The figure reveals that the corresponding values for the output probability  $\Pr(a \mid b, e)$

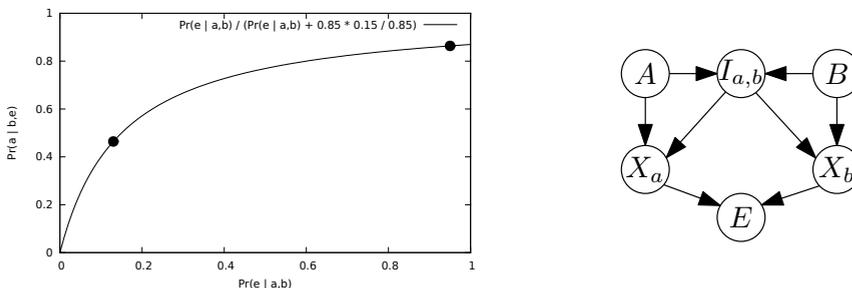


Figure 2: The sensitivity function describing  $\Pr(a \mid b, e)$  in terms of the noisy-OR calculated probability  $\Pr(e \mid a, b)$  (left). The conceptual model underlying the intercausal cancellation model (right).

can differ substantially, which can result in erroneous conclusions being drawn. From our investigations, we concluded that caution is advised with using the noisy-OR model especially for causal mechanisms in which the modelled causes exhibit at least some cancellation effect.

Upon describing the interaction effects among pharmaceutical substances in the real-world domain of pharmacology, we encountered various types of cancellation. These patterns of cancellation forestall the use of the noisy-OR model for specifying the conditional probability tables involved. For example, reducing the risk of osteoporosis can be achieved either by taking calcium supplements or through medication with bisphosphonates. Upon concurrent intake however, the effects of both will be decreased and possibly negated. The noisy-OR model would however define an increased overall effect of the two substances for the conditional probability table for the variable modelling osteoporosis, which would result in counterintuitive reasoning patterns upon propagation. For developing our Bayesian network in pharmacology, it would thus be highly advantageous to have available a causal interaction model to help describe cancellation effects. We expect that such a model will find wider use in a range of biomedical, chemical and environmental domains.

## 4 Designing an Intercausal Cancellation Model

Based upon possible patterns of cancellation among the causes of a common effect, we define a new causal interaction model. In Section 4.1 we begin by studying the intercausal effects in the conditional probability tables that result from reversing the noisy-OR model, and conclude that this model cannot be used to describe cancellation effects. In Section 4.2, we develop our new interaction model and show that this model does allow capturing different patterns of intercausal cancellation.

### 4.1 The reversed noisy-OR model

Our intercausal cancellation model is aimed at application to causal mechanisms in which the modelled causes serve to annihilate an effect that is present apriori. Since the noisy-OR model essentially embeds a logical OR, it is intended for a reversed mechanism: the noisy-OR model assumes the common effect  $e$  of a causal mechanism to be absent apriori and assumes each cause in itself to basically give rise to the effect. As a first step in the design of an interaction model describing cancellation effects, we reverse the conditional probability table which results from application of the noisy-OR model and study its (qualitative) properties.

We consider a causal mechanism with the binary effect variable  $F$  and the two binary cause variables  $A$  and  $B$ ; each of the causes  $a$  and  $b$  in essence serves to give rise to the effect  $f$ . By applying the noisy-OR model to this mechanism, we establish for the conditional probability table of  $F$  the probabilities  $\Pr(f | \bar{a}, b) = 1 - x$  and  $\Pr(f | a, \bar{b}) = 1 - y$  for some (small) values  $x, y \in [0, 1]$ ; the probability  $\Pr(f | \bar{a}, \bar{b})$  is set to zero and the probability  $\Pr(f | a, b)$  is computed to be  $1 - x \cdot y$ . To investigate application of the noisy-OR model to a causal mechanism in which the modelled causes serve to annihilate the effect, we introduce a new effect variable  $E$  with  $E = \bar{F}$ . For the conditional probability table of  $E$ , we now find:

$$\begin{aligned} \Pr(e | \bar{a}, \bar{b}) &= 1 - \Pr(f | \bar{a}, \bar{b}) = 1 \\ \Pr(e | \bar{a}, b) &= 1 - \Pr(f | \bar{a}, b) = x \\ \Pr(e | a, \bar{b}) &= 1 - \Pr(f | a, \bar{b}) = y \\ \Pr(e | a, b) &= 1 - \Pr(f | a, b) = x \cdot y \end{aligned}$$

We observe that the computed conditional probability table indeed captures the knowledge that in the absence of both causes, the effect  $e$  is present; the causes  $a$  and  $b$  moreover, each serve to annihilate the effect to some extent. We further find that the combined effect of  $a$  and  $b$  on the probability of  $e$  is stronger than the effect of each of the two causes separately, since  $x \cdot y \leq \min\{x, y\}$ . From an intercausal cancellation model however, we would wish to find that  $\Pr(e | a, b) \geq \min\{x, y\}$ . We conclude that the noisy-OR model cannot be used to capture patterns of cancellation, not even by reversing the modelled causal effects.

Upon studying the properties of qualitative probability embedded in the conditional probability table computed above, we find negative qualitative influences of both  $A$  and  $B$  on  $E$ . From an intercausal cancellation model with  $\Pr(e | a, b) = \min\{x, y\}$ , we would equally have negative qualitative influences of both  $A$  and  $B$  on  $E$ . With  $\Pr(e | a, b) > \min\{x, y\}$  however, we should find non-monotonic influences. We further observe that the table above embeds a positive additive synergy of  $A$  and  $B$  on  $E$ , since  $1 + x \cdot y \geq x + y$ . From an intercausal annihilation model, with  $\Pr(e | a, b) \geq \min\{x, y\}$ , a positive additive synergy is indeed what we would wish to find. For the final properties, we consider the two product synergies embedded in the computed table. Given  $e$ , the table reveals a zero product synergy; given  $\bar{e}$ , we find a negative product

synergy from  $0 \leq (1-x) \cdot (1-y)$ . From an intercausal cancellation model however, we would wish to find a positive product synergy in the presence of the effect variable  $E$ : informally speaking, when  $e$  is observed, either both causes must be absent or the influence of cause  $a$  on  $e$  must have been cancelled out to at least some extent by the presence  $b$  of the other cause. We further would wish to find a negative product synergy in the absence of the effect: given  $\bar{e}$ , the presence of the one cause must induce a negative influence on the other cause variable, just as with the reversed noisy-OR model.

## 4.2 A conceptual model of cancellation

We recall that the noisy-OR model has an underlying conceptual model that associates with each cause variable an inhibitor variable to describe the uncertainty of its causal effect. We also recall that the model's function for computing the joint effect of multiple simultaneous causes builds upon an independence assumption for these inhibitor variables. For the design of our intercausal cancellation model, we now use the same underlying idea and develop a conceptual model with inhibitor variables. In this model, the inhibitor variables describe the cancellation effects among the various causes and cannot be assumed independent.

The conceptual model used for defining our intercausal cancellation model is depicted in Figure 2. The causal mechanism under study includes again the two cause variables  $A$  and  $B$  and the effect variable  $E$ . The two intermediate variables  $X_a$  and  $X_b$  represent the hidden processes through which the causes  $a$  and  $b$  exert their annihilation influence on the common effect  $e$ . The inhibitor variable  $I_{a,b}$  in essence serves to capture the cancellation effect among the causes  $a$  and  $b$ . We now first specify the conditional probability tables for the effect variable  $E$  and for the inhibitor variable  $I_{a,b}$ :

$$\begin{array}{llll} \Pr(i_{a,b} | \bar{a}, \bar{b}) = 0 & \Pr(\bar{i}_{a,b} | \bar{a}, \bar{b}) = 1 & \Pr(e | \bar{x}_a, \bar{x}_b) = 1 & \Pr(\bar{e} | \bar{x}_a, \bar{x}_b) = 0 \\ \Pr(i_{a,b} | a, \bar{b}) = 0 & \Pr(\bar{i}_{a,b} | a, \bar{b}) = 1 & \Pr(e | x_a, \bar{x}_b) = 0 & \Pr(\bar{e} | x_a, \bar{x}_b) = 1 \\ \Pr(i_{a,b} | \bar{a}, b) = 0 & \Pr(\bar{i}_{a,b} | \bar{a}, b) = 1 & \Pr(e | \bar{x}_a, x_b) = 0 & \Pr(\bar{e} | \bar{x}_a, x_b) = 1 \\ \Pr(i_{a,b} | a, b) = 1 & \Pr(\bar{i}_{a,b} | a, b) = 0 & \Pr(e | x_a, x_b) = 0 & \Pr(\bar{e} | x_a, x_b) = 1 \end{array}$$

We note that these probability tables express the information that the inhibitor is activated only if both causes are present. Each of the processes  $x_a$  and  $x_b$  suffices to annihilate the effect  $e$  which is a priori present.

For defining the conditional probability tables for the process variables  $X_a$  and  $X_b$ , we observe that an annihilation process is initiated only by its associated cause; there is however a small probability that the cause will not be able to evoke the process. If a cancellation effect is induced, expressed by  $i_{a,b}$ , then the probability of the process not occurring is increased. The tables for the process variables are now defined as:

$$\begin{array}{llll} \Pr(x_a | \bar{a}, \bar{i}_{a,b}) = 0 & \Pr(\bar{x}_a | \bar{a}, \bar{i}_{a,b}) = 1 & \Pr(x_b | \bar{b}, \bar{i}_{a,b}) = 0 & \Pr(\bar{x}_b | \bar{b}, \bar{i}_{a,b}) = 1 \\ \Pr(x_a | \bar{a}, i_{a,b}) = 0 & \Pr(\bar{x}_a | \bar{a}, i_{a,b}) = 1 & \Pr(x_b | \bar{b}, i_{a,b}) = 0 & \Pr(\bar{x}_b | \bar{b}, i_{a,b}) = 1 \\ \Pr(x_a | a, \bar{i}_{a,b}) = 1 - y & \Pr(\bar{x}_a | a, \bar{i}_{a,b}) = y & \Pr(x_b | b, \bar{i}_{a,b}) = 1 - x & \Pr(\bar{x}_b | b, \bar{i}_{a,b}) = x \\ \Pr(x_a | a, i_{a,b}) < 1 - y & \Pr(\bar{x}_a | a, i_{a,b}) > y & \Pr(x_b | b, i_{a,b}) < 1 - x & \Pr(\bar{x}_b | b, i_{a,b}) > x \end{array}$$

In these probability tables, the values  $x$  and  $y$  match the conditional probabilities  $\Pr(e | \bar{a}, b) = x$  and  $\Pr(e | a, \bar{b}) = y$  for the causal mechanism at hand. These two probabilities are parameter probabilities of our intercausal cancellation model, just as for the noisy-OR model; the two probabilities thus have to be assessed explicitly by domain experts or be retrieved from data.

In the probability tables above for  $X_a$  and  $X_b$  no values have been provided as yet for  $\Pr(x_a | a, i_{a,b})$  and  $\Pr(x_b | b, i_{a,b})$ . The indicated constraints are derived from the informal meanings of the process and inhibitor variables: upon activation of the inhibitor  $i_{a,b}$ , the processes  $x_a$  and  $x_b$  are less likely to operate than in the inhibitor's absence. To arrive at fully specified probability tables for  $X_a$  and  $X_b$ , we now choose

$$\begin{array}{l} \Pr(\bar{x}_a | a, i_{a,b}) = y + \alpha_a \cdot (1 - y) \\ \Pr(\bar{x}_b | b, i_{a,b}) = x + \alpha_b \cdot (1 - x) \end{array}$$

with  $0 \leq \alpha_a, \alpha_b \leq 1$ ; note that the formulas are guaranteed to yield valid probability values. The degree of cancellation among the two causes is thus regulated by the parameters  $\alpha_a$  and  $\alpha_b$ . For example, for the parameter  $\alpha_a$  we find that  $\alpha_a = 0$  implies  $\Pr(\bar{x}_a | a, i_{a,b}) = y$  which expresses that the influential effect of the cause  $a$  on  $e$  is not cancelled out at all;  $\alpha_a = 1$ , and hence  $\Pr(\bar{x}_a | a, i_{a,b}) = 1$ , describes full cancellation of the effect of  $a$ . Any choice with  $0 < \alpha_a < 1$  captures a limited or partial cancellation effect: the closer  $\alpha_a$  is to 1, the stronger the cancellation effect of  $b$  on the influence of  $a$  will be.

Using the conceptual model developed above, we now derive the conditional probability  $\Pr(e | a, b)$  for our causal mechanism, and thereby complete the specification of our intercausal cancellation model. By

building upon independence properties and applying well-known rules of probability, we find that

$$\begin{aligned}\Pr(e \mid a, b) &= \Pr(e \mid \bar{x}_a, \bar{x}_b) \cdot \Pr(\bar{x}_a, \bar{x}_b \mid a, b, i_{a,b}) \\ &= \Pr(\bar{x}_a \mid a, i_{a,b}) \cdot \Pr(\bar{x}_b \mid b, i_{a,b}) \\ &= (y + \alpha_a \cdot (1 - y)) \cdot (x + \alpha_b \cdot (1 - x))\end{aligned}$$

Upon applying the cancellation model for a real-world domain, the Bayesian-network engineer should choose appropriate values for the parameters  $\alpha_a$  and  $\alpha_b$ . We will argue in Section 5 that an in-depth discussion, with a domain expert, of the annihilation processes induced by the modelled causes, will provide quite some insight in the cancellation effects to be captured. When the discussion reveals that causes exhibit (mutual or one-sided) full cancellation of their annihilating influences, the choice of parameter values is readily made. For modelling partial cancellation effects however, appropriate values need be assessed for  $\alpha_a$  and  $\alpha_b$ . To alleviate the elicitation burden involved, we propose to use a maximum-entropy choice of value for the two parameters alike. With  $\alpha = \alpha_a = \alpha_b = \frac{1}{2}$ , we find that

$$\begin{aligned}\Pr(e \mid a, b) &= (y + \alpha \cdot (1 - y)) \cdot (x + \alpha \cdot (1 - x)) \\ &= \frac{1}{4} \cdot (1 + x + y + x \cdot y)\end{aligned}$$

With  $\alpha = \frac{1}{2}$ , the conditional probability  $\Pr(e \mid a, b)$  thus takes its value from the interval  $[\frac{1}{4}; 1]$ .

To summarise, our intercausal cancellation model for the causal mechanism from Section 2, now defines the conditional probability table for the effect variable  $E$  to be

$$\begin{aligned}\Pr(e \mid \bar{a}, \bar{b}) &= 1 \\ \Pr(e \mid a, \bar{b}) &= y \\ \Pr(e \mid \bar{a}, b) &= x \\ \Pr(e \mid a, b) &= \frac{1}{4} (1 + x + y + x \cdot y)\end{aligned}$$

We note that the above table was constructed with the maximum-entropy choice of value for the parameters regulating the cancellation effects. If sufficient insights are available of the strengths of the cancellation effects, then more appropriate parameter values can be elicited and used with the model.

### 4.3 Properties of the cancellation model

In Section 4.1, we argued that the intercausal cancellation model being developed should embed particular properties of qualitative probability in the probability tables constructed from its application. We now verify that these properties are indeed induced by our model. Upon doing so, we assume the maximum-entropy values for the parameters  $\alpha_a$  and  $\alpha_b$ . It is readily verified that analogous results are found for other values.

We begin by investigating the qualitative influences of the two cause variables on the effect variable. For the qualitative influence of the variable  $A$  on the variable  $E$ , we establish that  $\Pr(e \mid a, \bar{b}) < \Pr(e \mid \bar{a}, \bar{b})$ , that is, in the absence of the cause  $b$ , observing the value  $a$  serves to decrease the probability of the effect  $e$  being present. The sign of the difference  $\Pr(e \mid a, b) - \Pr(e \mid \bar{a}, b)$  on the other hand, is dependent of the exact probability values  $x$  and  $y$ ; for example, with  $x = 1$  and  $y < 1$  the difference is negative, while with  $x = 0$  it is positive. The influence of  $A$  on  $E$  will thus be either negative or non-monotonic, but can never be positive. Similar observations hold for the qualitative influence of the cause variable  $B$  on  $E$ . For the additive synergy exhibited by the two cause variables, we find from the induced probability table that

$$\begin{aligned}\Pr(e \mid a, b) + \Pr(e \mid \bar{a}, \bar{b}) - \Pr(e \mid \bar{a}, b) - \Pr(e \mid a, \bar{b}) &= \\ &= \frac{1}{4} \cdot (1 + x + y + x \cdot y) + 1 - x - y \geq 0\end{aligned}$$

We conclude that the cause variables  $A$  and  $B$  exhibit a positive additive synergy upon the effect variable  $E$ . With respect to the product synergy between the two cause variables, we derive the signs of the intercausal influences in the presence and in the absence of the effect, respectively. Given the observed presence of the effect, we find for the product synergy of  $A$  and  $B$  that

$$\begin{aligned}\Pr(e \mid a, b) \cdot \Pr(e \mid \bar{a}, \bar{b}) - \Pr(e \mid \bar{a}, b) \cdot \Pr(e \mid a, \bar{b}) &= \\ &= \frac{1}{4} \cdot (1 + x + y + x \cdot y) - x \cdot y \\ &= \frac{1}{4} \cdot (1 + x + y - 3 \cdot x \cdot y) \geq 0\end{aligned}$$

Given  $e$  therefore, a positive product synergy is found. Given  $\bar{e}$ , we find a negative product synergy:

$$\begin{aligned} \Pr(\bar{e} \mid a, b) \cdot \Pr(\bar{e} \mid \bar{a}, \bar{b}) - \Pr(\bar{e} \mid \bar{a}, b) \cdot \Pr(\bar{e} \mid a, \bar{b}) &= \\ &= -(1-x) \cdot (1-y) \leq 0 \end{aligned}$$

From the above observations, we conclude that the conditional probability table constructed with the cancellation model indeed embeds the properties of qualitative probability that capture intercausal cancellation.

## 5 Real-life examples of intercausal cancellation

To investigate the practicability of the intercausal cancellation model designed above, we studied several examples from the domain of pharmacology with the help of a domain expert. Each example pertained to a specific condition and involved two or more pharmaceutical substances expected to have therapeutic effects on the condition. We asked the expert to assess conditional probabilities for the condition being present given all possible treatment regimes.

**Example 1.** We consider two different treatments administered to patients suffering from epigastric pains as a result of pyrosis, that is, from heartburn associated with regurgitation of gastric acid. Pyrosis-associated pains are relieved by the administration of either antacids or proton pump inhibitors. Antacids essentially have a neutralisation reaction and reduce the acidity of the stomach contents. Proton pump inhibitors work not on the stomach contents itself but on the cells that line the stomach and inhibit the production of acid by these cells. Upon concurrent intake the proton pump inhibitors bar the production of acids, with a reasonably pH-neutral stomach contents for a result. The neutralisation effect of the antacids then no longer contributes to pain relief. Based upon knowledge of these processes, we concluded that the interaction among the two substances constitutes a single-sided, possibly full cancellation. The domain expert provided the following assessments for the conditional probability table of the effect variable  $E$  modelling epigastric pains:

$$\begin{aligned} \Pr(e \mid \bar{a}, p) &= 0.05 \\ \Pr(e \mid a, \bar{p}) &= 0.70 \\ \Pr(e \mid a, p) &= 0.05 \end{aligned}$$

where  $a$  models the administration of antacids and  $p$  captures medication by proton pump inhibitors. We note that these probabilities support the conclusion of a single-sided full cancellation effect. We further note that use of our intercausal cancellation model, with  $\alpha_a = 1$  and  $\alpha_p = 0$  to describe the cancellation effects, would have resulted in the same conditional probability table. For comparison purposes, we would like to mention that the reversed noisy-OR model would have yielded the probability  $\Pr(e \mid a, p) = 0.04$ .  $\square$

**Example 2.** We now consider two possible treatments for patients with primary type-1 osteoporosis. In healthy persons, the amount of bone mass in the skeleton is controlled by two types of cell: the osteoclasts break down, or resorb, bone material and the osteoblasts form bone tissue from calcium. With osteoporosis, the net rate of bone resorption exceeds the rate of bone formation, resulting in a decrease in bone mass. The risks of bone fracture typically associated with osteoporosis, are reduced by calcium supplementation and through medication with bisphosphonates. Calcium supplements are aimed at providing the osteoblasts with sufficient material for bone formation. Bisphosphonates on the other hand inhibit the resorption of bone by binding to the calcium in the bone tissue to increase osteoclast death rate. Bisphosphonates are provided in small dosage but are much more effective than calcium supplementation. Upon concurrent intake however, the bisphosphonates will bind to the calcium supplements in the stomach rather than to the calcium in the bone tissue. As a result the therapeutic effects of both treatments are decreased. Because of their small dosage the effect of the bisphosphonates is likely to disappear altogether; since the calcium is administered in larger quantities, some effect of the supplementation is expected to remain. Based upon knowledge of the therapeutic processes involved, we concluded that the interaction among the two treatment regimes induces a pattern of cancellation in which the effect of the bisphosphonates is fully cancelled out and the effect of the calcium supplementation is partly cancelled. The domain expert provided the following assessments for the conditional probability table of the effect variable  $E$  modelling primary type-1 osteoporosis:

$$\begin{aligned} \Pr(e \mid \bar{b}, c) &= 0.85 \\ \Pr(e \mid b, \bar{c}) &= 0.15 \\ \Pr(e \mid b, c) &= 0.95 \end{aligned}$$

where  $b$  captures medication by bisphosphonates and  $c$  models the administration of calcium supplements. We note that the above conditional probabilities support our conclusions about the pattern of cancellation among the two substances. We further note that use of our intercausal cancellation model, with  $\alpha_b = 1$  and  $\alpha_c = \frac{1}{2}$ , would have resulted in the conditional probability  $\Pr(e \mid b, c) = 0.93$ . For comparison purposes, we would like to mention that the reversed noisy-OR model would have yielded the probability  $\Pr(e \mid b, c) = 0.13$ . The effect of this difference is visualised in the graph in Figure 2 on the left, with  $\Pr(a \mid b, e)$  for  $\Pr(b \mid e, c)$  and  $\Pr(e \mid a, b)$  for  $\Pr(e \mid b, c)$ . Given an apriori compliance of  $\Pr(b) = 0.85$  with regard to the bisphosphonates, the reversed noisy-OR model suggests that an elderly patient who is taking her calcium supplements and nonetheless has osteoporosis, would have taken her bisphosphonates with probability 0.46. In reality however, this probability is much higher, which is reflected by the probability 0.86 computed through the use of our cancellation model.  $\square$

## 6 Conclusions

While a variety of causal interaction models are available to Bayesian-network engineers, tailored models for describing cancellation effects among causes had not been designed as yet. In this paper, we proposed a new causal interaction model to describe such intercausal cancellation. The new model serves to lessen the burden of probability elicitation upon constructing a Bayesian network with the help of domain experts. It was designed from first principles, along the same lines as the popular noisy-OR model. The cancellation model even requires the same parameters as the noisy-OR model, yet with the possible addition of regulation parameters for which generally a default value can be assumed. The main advantage of the new model lies not so much in reducing the number of parameter probabilities required however, but in the observation that assessment is forestalled for the probability which is often considered the hardest to provide, that is, the probability given the presence of multiple simultaneous causes. Although we presented our model to apply to causal mechanisms with annihilating cause variables, it is also applicable by reversion to mechanisms in which the cause variables serve to cause the common effect, yet whose influences can be cancelled out. Our future research will focus on extending the cancellation model to accommodate more than two cause variables on the one hand and to allow the common effect to be initially absent with a small (leak) probability on the other hand. We expect that these and further extensions of the basic ideas presented in the current paper will result in a generally applicable cancellation model for ready use by network engineers.

## References

- [1] L.C. van der Gaag, S. Renooij, C.L.M. Witteman, B.M.P. Aleman, B.G. Taal. Probabilities for a probabilistic network: a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, pp. 123–148, 2002.
- [2] M. Henrion. Some practical issues in constructing belief networks. In: J. Lemmer, T. Levitt, L. Kanal (eds). *Proceedings of the Third Annual Conference on Uncertainty in Artificial Intelligence*, Elsevier, New York, NY, pp. 161–174, 1987.
- [3] F.J. Díez. Parameter adjustment in Bayes networks. The generalized noisy OR gate. In: D. Heckerman, A. Mamdani (eds). *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA, pp. 99–105, 1993.
- [4] P.J.F. Lucas. Bayesian network modelling through qualitative patterns. *Artificial Intelligence*, pp. 233–263, 2005.
- [5] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA, 1988.
- [6] A. Oniško, M.J. Druzdzel, H. Wasyluk. Learning Bayesian network parameters from small data sets: application of noisy-OR gates. *International Journal of Approximate Reasoning*, pp. 165–182, 2001.
- [7] J. Bolt, L.C. van der Gaag. An empirical study of the use of the noisy-OR model in a real-life Bayesian network. In: E. Hüllermeier, R. Kruse, F. Hoffmann (eds). *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Methods*, Springer, Berlin, pp. 11–20, 2010.
- [8] S.P.D. Woudenbergh, L.C. van der Gaag. Using the noisy-OR model can be harmful ... but it often is not. In: W. Liu (ed). *Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Springer, Heidelberg, pp. 122–133, 2011.

# Shortest Path Gaussian Kernels for State Action Graphs: An Empirical Study

Saba Yahyaa

Bernard Manderick

*Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels*

## Abstract

We approximate action-value functions by a linear combination of shortest path Gaussian kernels. These kernels are defined on the state graph and the state-action graph derived from the Markov decision problem that the agent has to solve. An empirical comparison on a testbed of 3 MDPs shows that these kernels work better than other basis functions, e.g. the smoothest eigenfunctions of the normalized Laplacian of the state graph and the state-action graph.

## 1 Introduction

For a given Markov decision problem (MDP), the action-value function <sup>1</sup>  $Q^\pi(s, a)$  gives the total expected discounted reward when the agent starts in state  $s$ , executes action  $a$  first and follows the policy  $\pi$  thereafter. The optimal policy  $\pi^*$  maximizes this reward and it can be derived, e.g. by policy iteration, from the optimal action-value function  $Q^*$  [12]. When the state space  $S$  and/or the action space  $A$  are large finite discrete spaces<sup>2</sup>, policy iteration becomes impractical due to the curse of dimensionality. One way to overcome this problem is to use approximate policy iteration of which representation policy iteration is a special case [14]. The action-value function  $Q^\pi(s, a)$  belongs to a high dimensional Hilbert space  $H$  with dimension  $|S \times A|$  and is approximated by  $\hat{Q}^\pi(s, a)$  which belongs to a lower dimensional subspace  $K$  spanned by a set of  $k$  predefined basis functions  $\phi_i(s, a)$ ,  $i = 1, \dots, k$ . Usually, the dimension of this subspace  $K$  is much smaller than the dimension of the original space  $H$ ,  $k \ll |S \times A|$ . Since we deal with Hilbert spaces, i.e. complete vector spaces with an inner product, there always exists a *unique best approximation*, the orthogonal projection onto that subspace  $K$ , but the quality of the approximation depends very much on the selected basis functions [2]. For instance, a given action-value function might be well approximated by a combination of radial basis functions while this is not possible by a combination of polynomials up to some degree.

Here, we assume that the agent does not have a model of its environment. As a result, the state-action value function  $Q^\pi$  is not known beforehand but has to be learned by the agent itself. As a result the weights  $w_i^\pi$  of the best approximation cannot be found by orthogonal projection. Fortunately, Lagoudakis and Parr [7] have introduced Least Squares Policy Iteration (LSPI) that learns these weights  $w_i^\pi$  from the agent's interaction with the environment.

Basis functions are defined on the state space or the state-action space provided with distance function. For many MDPs, we can define a 'Euclidean' distance on these spaces. For instance, for the grid problem shown in Figure 1, the states 2, 7 and 8 are close to state 1 but state 26 is far away. We face three potential problems when defining basis functions. First, the underlying MDP is not known to the agent and so it cannot determine the distance between states. Second, value functions might have 'discontinuities'<sup>3</sup> which make good approximation as a combination of 'smooth' basis functions difficult. For instance, states 7 and

<sup>1</sup>In this paper, we focus on the state-action value function  $Q^\pi(s, a)$  and we use the state value function  $V^\pi(s)$  for explanatory purposes only. Also, we use the term value function in generic way, it refers to either the state-action value function or the state value function depending on the context.

<sup>2</sup>The same argument applies to continuous state and/or action spaces but they are not considered here.

<sup>3</sup>The notions 'continuity' and 'smooth' only make sense for continuous domains but equivalent notions for discrete domains can be defined in terms of the Sobolev norm for discrete domains [14].

15 are close to each other according to the 'Euclidean' distance but are separated by an unaccessible region. As a result, the difference between the values of these states might be large resulting in a 'discontinuity' in the value function. And third, not every set of predefined basis functions guarantees good approximation of the value function.

To tackle these problems, Mahadevan and Maggioni proposed 1) to construct a state graph based on the experience obtained by the agent while exploring the state space, 2) to define a distance based on the shortest path between nodes in that graph, e.g. the shortest path between states 7 and 15 has length 6 but the Euclidean distance between these states is 8, and 3) to construct a basis derived from properties of that graph and not in an ad hoc manner [14]. Next, Sugiyama et al. [10] have introduced the shortest path Gaussian kernels and they have shown that basis consisting of such kernels outperforms the basis functions used by Mahadevan and Maggioni [14]. Finally, Osentoski has extended the state graph approach to state-action graphs [16].

In this paper, we extend shortest path Gaussian kernels to state-action graphs and we show experimentally on a testbed of MDPs that this basis gives better results than the ones used in related work [10, 14, 16].

The rest of the paper is organized as follows: In section 2 we present Markov decision problems, least squares policy iteration, representation policy iteration, and several variants of Gaussian kernels to be used as basis functions. In section 3, we give an overview of different kinds of bases for approximating value functions. In section 4, we describe the testbed of MDPs used in our comparison followed by our experimental results. Finally, we conclude and discuss future work.

## 2 Preliminaries

Here, we define MDPs, least squares policy iteration (LPSI), the representation policy iteration (RPI) framework introduced by Mahadevan and co-workers [14], and the family of Gaussian kernels used in our experiments.

A *finite and discounted MDP* is a 5-tuple  $(S, A, P, R, \gamma)$  where the state space  $S$  is a finite set of states, the action space  $A$  is a finite set of actions, the transition function  $P : S \times A \times S \rightarrow [0, 1] : (s, a, s') \rightarrow P(s, a, s')$  gives the conditional probability  $p(s'|s, a)$  that the environment transits to state  $s'$  when the agent takes action  $a$  in state  $s$ , the expected reward function  $R : S \times A \times S \rightarrow \mathbb{R} : (s, a) \rightarrow R(s, a)$  gives the immediate reward to be expected when the agent transits to state  $s'$  after taking action  $a$  in state  $s$ , and  $\gamma \in [0, 1)$  is the discount factor that determines the present value of future rewards [9, 12]. A deterministic policy is a mapping  $\pi : S \rightarrow A$  that determines which action  $a \in A$  the agent takes in each state  $s \in S$ . The state-action value function  $Q^\pi(s, a) : S \times A \rightarrow \mathbb{R}$  under policy  $\pi$  gives the total discounted expected reward  $\mathbb{E}_\pi(\sum_{i=t}^{\infty} \gamma^i r_i)$  when the agent starts in state  $s$ , takes action  $a$  and follows policy  $\pi$  thereafter. The goal of the agent is to find the policy  $\pi^*$  that maximizes the state-action value function for every state  $s$  and action  $a$ :  $\pi^*(s) = \operatorname{argmax}_{a \in A} Q^*(s, a)$  where  $Q^*(s, a) = \max_{\pi} Q^\pi(s, a)$  is the optimal state-action value function. If the MDP is completely known then algorithms such as policy iteration (PI) and value iteration (VI) will find the optimal policy  $\pi^*$  [12]. In case of finite MDPs there is always a deterministic optimal policy [9].

In practice, however, the agent does not know the transition function  $P$  and the expected reward function  $R$ . Instead, the agent must rely on information collected while interacting with the environment in order to learn the optimal policy. This information consists of trajectories (of length  $n$ ) of samples of the form  $(s_t, a_t, r_t, s_{t+1})$  or  $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$ , the subscript  $t$  indicates the time step at which the sample is taken. For finite MDPs, the state-action value function  $Q^\pi$  for any policy  $\pi$  can be represented by a lookup table of size  $|S \times A|$ , one entry per state-action pair. When the state and/or action space are large, this approach becomes computationally infeasible due to the curse of dimensionality and one has to rely on function approximation methods instead.

Lagoudakis and Parr [7] proposed *least square policy iteration (LSPI)* to approximate state-action value functions in order to find the optimal policy. For each policy  $\pi$ , LSPI approximates the state-action value function  $Q^\pi(s, a)$  by a linear combination  $\hat{Q}^\pi(s, a) = \sum_{i=1}^k \phi_i(s, a) w_i^\pi$  of  $k \ll |S \times A|$  predefined basis functions. The weight parameters  $(w_i^\pi)_{i=1}^k$  are estimated. LSPI begins with an initial policy and approximates the corresponding state-action value function. Then LSPI iteratively performs two steps: policy evaluation which approximates the state-action value function given the current policy and policy improvement which adapts the current policy as long as improvement is possible. The predefined basis functions determine whether good approximations are obtained or not.

Mahadevan and co-workers introduced the representation policy iteration (RPI) framework where the basis functions are learned instead of predefined and as a consequence better approximation might be obtained [14]. These basis functions are derived from properties of the state graph (or the state-action graph) constructed during this learning process. RPI consists of 3 iterated steps. First, trajectories of samples are collected while the agent is following an initial policy, e.g. a random walk. These trajectories are used to construct either the state graph (or the state-action graph). States (or state-action pairs) are connected in the corresponding graph when they form temporally successive states (or state-action pairs), i.e. the states  $s_t$  and  $s_{t+1}$  will be connected in the state graph (or the state-action pairs  $(s_t, a_t)$  and  $(s_{t+1}, a_{t+1})$  will be connected in the state-action graph). We also consider weighted graphs: when nodes are connected a weight is assigned that gives the relative frequency of transition between these 2 nodes and this weight is derived from the trajectory of samples used. Next, a basis is constructed using properties of these graphs, e.g. the smoothest eigenfunctions of the normalized Laplacian of that graph are used as basis functions. This is the representation learning step. Finally, the best policy that can be written as a linear combination of the basis functions is estimated using LSPI. This is the control learning step. More details about LSPI and RPI can be found in the papers [7, 14].

In our experiments, we compare a family of *Gaussian kernels* of the form  $\phi(s) = e^{-\frac{d^2(c,s)}{2\sigma^2}}$ . They differ in the distance  $d$  used. Here,  $c$  is the center of Gaussian kernel,  $\sigma$  is the standard deviation that controls the spread around the center, and  $d$  is either the Euclidean distance, a distance based on the shortest path between nodes in the state graph (or the state-action graph) or a combination of both. A basis consists of  $k$  Gaussian kernels of the form  $\phi_i(s) = e^{-d^2(c_i,s)/2\sigma^2}$ ,  $i = 1, \dots, k$ , where  $\sigma$  is the common standard deviation. Both the centers and the standard deviation have to be tuned for the problem at hand.

The family of Gaussian kernels considered in this paper are 1) Geodesic Gaussian Kernels GGKs( $SA$ ) basis functions defined over the state-action graph. They are an extension of Geodesic Gaussian kernels GGKs( $S$ ), introduced by Sugiyama et al. [10], defined on the state graph, and 2) Gaussian kernels functions GKBs-ESP defined in terms of the Euclidean distance and the shortest path distance. They are a modification of the functions introduced by Jakab [4] so that they can be used in off-line LSPI.

GGKs( $SA$ ) take the form  $\phi_i(s, a) = e^{-SP^2((s,a),c_i)/2\sigma^2}$  where  $SP((s, a), c_i)$  is the shortest path from the state-action pair  $(s, a)$  to the state-action pair represented by the center  $c_i$  of the corresponding basis function, and one of centers  $c_i$  must be a goal node<sup>4</sup>. The shortest path can be calculated efficiently using Dijkstra's algorithm, cf. Cormen et al. [17]. GGKs( $SA$ ) are smooth functions and well suited to approximate state-action value functions.

GKBs-ESP take the form  $\phi_i(s) = e^{-T_{SP}(s,c_i)/2\sigma^2}$  and combine the distance between nodes in the graph, i.e. the shortest path between these nodes, with the Euclidean distance between nodes in state space (or the state-action space). This combined distance is called the total shortest path  $T_{SP}$  and is calculated as follows:

$$T_{SP}^2(s, c_i) = d_{ED}^2(s, c_i^*) + SP^2(c_i^*, c_j^*) + d_{ED}^2(c_j^*, c_i)$$

where  $c_i^* = \operatorname{argmax}_{x=1\dots K} d_{ED}(s, c_x)$  and  $c_j^* = \operatorname{argmax}_{x=1\dots K} d_{ED}(c_i, c_x)$  are nodes in the state graph (or the state-action graph) and  $d_{ED}(s_1, s_2)$  is the Euclidean distance on the state space (or the state-action space).

### 3 Related Work

Below we discuss the basis functions considered in our experimental comparison. Compared to related work, we look at other basis functions defined on the state space and we also use basis functions defined on the state-graph and the state-action graph.

*State space:* The most common basis functions on the state space are polynomials and radial basis functions. Polynomials are global, easy to construct, and in each state  $s$ , they are spanned by the basis  $\{1, s, s^2, \dots, s^{k-1}\}$ . However, it proved difficult to approximate smooth functions [7]. Radial basis functions (RBFs) are a special case of the Gaussian kernels defined in section 2. They are local functions centered around a selected state.

*State graph:* As explained in section 2, we first build the state graph  $G_S$  using trajectories of samples obtained by the agent while exploring the environment. Next, several matrices that reflect properties of that graph are constructed. Examples are the adjacency graph  $A_S$ , the combinatorial Laplacian  $L_S$ , the normalized Laplacian  $\mathcal{L}_S$  and the random walk matrix  $P_S$ . Third, spectral analysis is applied to these matrices, i.e.

<sup>4</sup>For GGKs, Sugiyama et al. [10] suggest one of the center node must be a goal node.

the eigenvalues and eigenvectors are determined [3]. Often, the result is a complete orthonormal basis of eigenvectors. These eigenvectors are interpreted as basis functions, called proto-value functions (PVFs), and the  $k$  smoothest ones are used to approximate value functions [14]. However, since these basis functions are globally smooth, they approximate poorly functions that are locally smooth or with different smoothness in different regions. More details can be found in the papers [13, 14]. *Wavelet Basis Functions* (WBFs) extend wavelet and multi-scale analysis of functions defined on Euclidean spaces to functions defined on graphs. WBFs are better to approximate value functions that have discontinuities and/or have different degree of smoothness, cf. [11, 15] for more details. Finally, *Shortest Path Gaussian Kernels*, introduced by Sugiyama et al. [10] and abbreviated as GGKs<sup>5</sup>, are smooth and use the topology of the state graph  $G_S$  induced by an MDP to approximate discontinuities in the original state space  $S$ . The definition domain of basis functions on the state graph is the set of states  $S$ . Since we want to approximate state-action value functions, we have to extend the definition domain of these basis functions to the set of state-action pairs  $S \times A$ , cf. [13] for more details about how this can be done.

*State-action graph*: Osentoski [16] extended the analysis based on the state graph  $G_S$  to the state-action graph  $G_{SA}$ . This time, the adjacency state-action matrix  $A_{SA}$ , the combinatorial Laplacian  $L_{SA}$ , the normalized Laplacian  $\mathcal{L}_{SA}$  and random walk matrix  $W_{SA}$  are derived from the state-action graph instead of the state graph. Then, spectral analysis is applied to these matrices to get the corresponding basis functions. This approach is harder from a computational point of view since the state graph contains  $|S|$  nodes while the state-action graph contains  $|S \times A|$  nodes but it has one important advantage. The basis functions that we extended from the set of states  $S$  to the set of state-action pairs  $S \times A$  are now adapted to actions available in each state. Osentoski [16] uses the  $k$  smoothest basis functions to approximate state-action value functions.

## 4 Experimental Comparison

First, we describe the test domain and the experimental setup. Then, we evaluate the performance of the basis functions on the state space, the state graph and the state-action graph discussed above. All experiments are implemented in MATLAB.

### 4.1 Test Domain and Experimental Setup

The *test domain* consists of 3 MDPs each with discount factor  $\gamma = 0.9$ . The 20 and 50-chain problems consist of a sequence of 20 and 50 states, respectively, labeled from  $s_1$  to  $s_{20}$  and from  $s_1$  to  $s_{50}$  [14]. In each state, the agent has 2 actions, either *Go Right* or *Go Left*. The actions succeed with probability 0.9 changing the state in the intended direction and fail with probability 0.1 changing the state in the opposite direction. For the 20-chain problem, the agent is rewarded 1 in states  $s_1$  and  $s_{20}$ , and 0 elsewhere. The optimal policy is *Go Left* from states  $s_1$  till  $s_{10}$  and *Go Right* from states  $s_{11}$  till  $s_{20}$ . For the 50-chain problem, the agent gets reward 1 in states  $s_{10}$  and  $s_{41}$  and 0 elsewhere. The optimal policy is to *Go Right* from state  $s_1$  till state  $s_{10}$  and from state  $s_{26}$  till state  $s_{40}$ , and to *Go Left* from state  $s_{11}$  till state  $s_{25}$  and from state  $s_{41}$  till state  $s_{50}$ . The third MDP, the grid world, is a simplification of the one used in [10]. The agent has 4 actions *Go Up*, *Down*, *Left* and *Right* and for each actions it transits to the intended state with probability 1. The agent gets reward 1 if it reaches the goal state,  $-1$  if it hits the wall, and 0 elsewhere. The state space together with the optimal policy are shown in Figure 1.

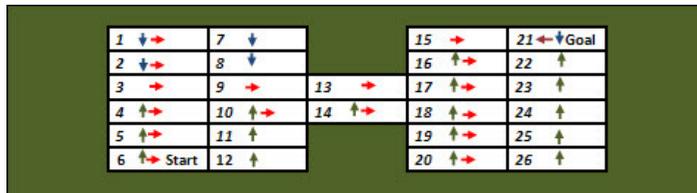


Figure 1: The grid world with 26 accessible states numbered from 1 to 26. The arrows show the optimal actions in each state.

<sup>5</sup>Sugiyama et al. use the name Geodesic Gaussian Kernel but since the distance used is based on shortest paths in a graph, we prefer shortest path Gaussian kernel instead but we abbreviate it as CGKs.

The *experimental setup* is as follows. For each of the 3 MDPs, we investigated which bases resulted in fast convergence in LSPI and found the optimal policy with a high probability. Therefore, we ran 10 experiments and determined 1) the relative frequency that the optimal policy was found, and 2) the average number of iterations needed by LSPI to converge to a policy, either the optimal policy or a suboptimal one. We used the same number of basis functions  $k$  as reported in related work. For Gaussian kernels the optimal standard deviation  $\sigma$  and the optimal position of the centers were determined beforehand. Often the centers were more or less equally spread over the state space  $S$  and the corresponding optimal standard deviation  $\sigma$  is about half the distance between consecutive centers, e.g. in case of the 20-chain problem with 5 kernels the centers were at the states  $s_2, s_5, s_{12}, s_{17}, s_{19}$  and the standard deviation  $\sigma = 2.5$ . The state graph and state-action graph were built using trajectories of 5,000 samples for the state graph and 5,000 times the number of actions available in each state for the state-action graph, i.e. a trajectory of 10,000 samples in case of the 20- and 50-chain problems since in each state we have 2 possible actions, and a trajectory of 20,000 samples for the grid world problem since there are 4 actions per state. For state graph [13], the combinatorial Laplacian  $L_S$ , and the normalized Laplacian  $\mathcal{L}_S$  have the same eigenvectors and the eigenvectors of the random walk  $P_S$  are related to  $\mathcal{L}_S$ , therefore, we took in the comparison only the eigenvectors of  $\mathcal{L}_S$  [13]. For state-action graph [16], the combinatorial Laplacian  $L_{SA}$ , and the normalized Laplacian  $\mathcal{L}_{SA}$  have the same eigenvectors, therefore, we took in the comparison only the eigenvectors of  $\mathcal{L}_{SA}$  [16].

## 4.2 Experimental Results

For the *20-chain problem*,  $k = 5$  basis functions are sufficient to get convergence except for polynomial basis functions and so they are not included in the comparison. The initial policy used is *Go Right* in every state of the chain. The basis functions are defined over the state space the optimal centers for the Gaussian kernels (RBFs in the text) using the Euclidean distance are  $\{s_2, s_5, s_{12}, s_{17}, s_{19}\}$ , and the optimal standard deviation is  $\sigma = 2.5$  which are the same for Gaussian kernels over state graph. For the state-action graph we get the following results depending on the Gaussian kernel used:

GGKs( $A_{SA}$ ) for state-action graphs: the optimal centers are located at  $\{sa_1, sa_{14}, sa_{23}, sa_{32}, sa_{40}\}$  and the optimal standard deviation is  $\sigma = 6.5$ .

GGKs( $W_{SA}$ ) for weighted state-action graphs: the optimal centers are located at  $\{sa_1, sa_{10}, sa_{20}, sa_{30}, sa_{40}\}$  and the optimal standard deviation is  $\sigma = 2.5$ .

GKBs-ESP( $W_{SA}$ ) for weighted state-action graphs: the optimal centers are located at  $\{sa_1, sa_{10}, sa_{20}, sa_{30}, sa_{40}\}$  and the optimal standard deviation is  $\sigma = 5.5$ .

Table 1 compares the different basis functions. It gives the relative frequency that the optimal policy was found and the average number of iterations that each set of basis functions needs to get convergence. Both figures are based on 10 runs.

Table 1: Performance on the 20-chain problem with  $k = 5$  basis functions. The column *Basis* gives the basis used while the columns *Freq.* and *Iters.* give the relative frequency that the optimal policy was found during 10 runs and the average number of iterations needed to converge to either an optimal or suboptimal policy.

Basis	Freq.	Iters.	Basis	Freq.	Iters.
RBFs	0.935	6.1	GKBs-ESP( $A_S$ )	0.995	4.2
$A_S$ -eigenbasis	0.485	2	$\mathcal{L}_{SA}$ -eigenbasis	0.95	5.9
$\mathcal{L}_S$ -eigenbasis	0.98	2.8	GGKs( $A_{SA}$ )	1	8.9
WBFs	0.9	2	GGKs( $W_{SA}$ )	1	7.1
GGKs( $A_S$ )	0.95	6.1	GKBs-ESP( $W_{SA}$ )	1	5.4

For the *50-chain problem* with the initial policy used *Go Right* in every state of the chain, firstly, we are going to explain how does the number of basis functions,  $k$  effects to find the optimal policy using the shortest path Gaussian kernels function GGKs( $A_S$ ) and the hybrid kernels GKBs-ESP( $A_S$ ). The comparison of state space basis was given in [8] and the comparison between the state space basis functions and state graphs basis function was given in [15].

For  $k = 5$  and GGKs( $A_S$ ), the optimal centers are located at  $\{s_{10}, s_{12}, s_{25}, s_{38}, s_{40}\}$ , the optimal standard deviation is  $\sigma = 3.75$ , convergence to the optimal policy happens with frequency 0.9 and takes 4

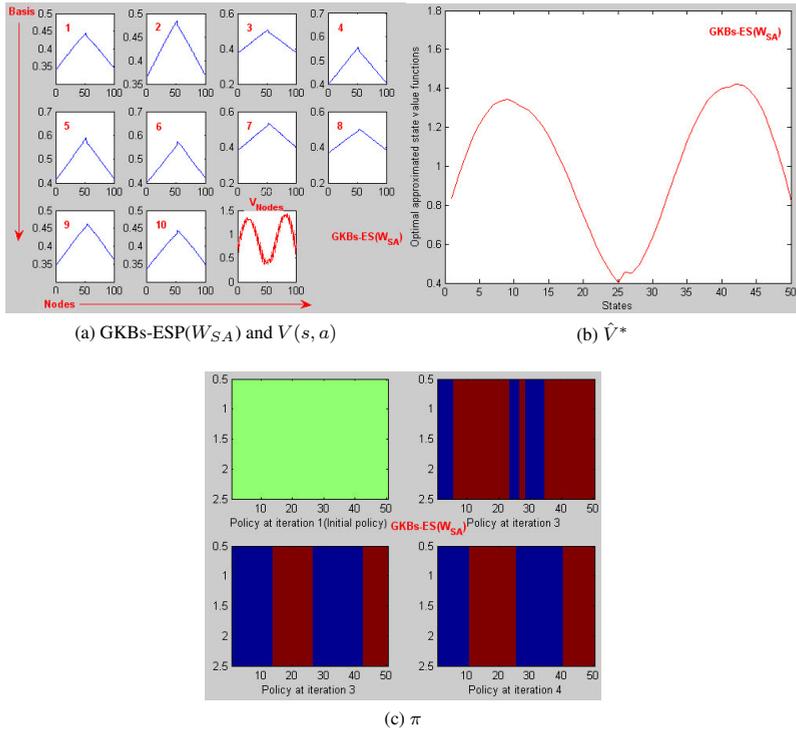


Figure 2: Performance of  $k = 10$  Gaussian kernels GKBs-ESP( $W_{SA}$ ) defined on the weighted state-action graph. Subfigure (a) shows the 10 basis functions, labeled from 1 to 10, followed by the estimated state-action value functions in terms of this basis. Subfigure (b) shows the approximation  $\hat{Q}^*$  of the optimal state-value function. And, subfigure (c) shows in each state the policy learned: *Go Right* in blue and *Go Left* in red.

iterations, while the optimal centers for GKBs-ESP( $A_S$ ) are located at  $\{s_6, s_{10}, s_{26}, s_{40}, s_{45}\}$ , the optimal standard deviation is  $\sigma = 3.5$ , convergence to the optimal policy happens with frequency 0.96 and takes 3 iterations.

For  $k = 10$  and GGKs( $A_S$ ), the optimal centers are located at  $\{s_5, s_{10}, s_{15}, s_{20}, s_{21}, s_{30}, s_{31}, s_{39}, s_{40}, s_{45}\}$ , the optimal standard deviation is  $\sigma = 2.75$ , convergence to the optimal policy happens with frequency 0.98 and takes 7 iterations, while the optimal centers for GKBs-ESP( $A_S$ ) are located at  $\{s_6, s_{13}, s_{17}, s_{25}, s_{26}, s_{28}, s_{30}, s_{38}, s_{41}, s_{48}\}$  and the optimal standard deviation is  $\sigma = 3.76$ , convergence to the optimal policy happens with frequency happen and takes 7 iterations.

For  $k = 15$  and GGKs( $A_S$ ), the optimal centers are located at  $\{s_5, s_9, s_{10}, s_{12}, s_{14}, s_{20}, s_{25}, s_{28}, s_{35}, s_{38}, s_{39}, s_{40}, s_{42}, s_{45}, s_{50}\}$ , the optimal standard deviation is  $\sigma = 2.75$ , convergence to the optimal policy always happens and takes 9 iterations, while the optimal centers for GKBs-ESP( $A_S$ ) are located at centers  $\{s_2, s_6, s_{11}, s_{16}, s_{19}, s_{21}, s_{23}, s_{27}, s_{29}, s_{32}, s_{35}, s_{38}, s_{43}, s_{45}, s_{49}\}$ , the optimal standard deviation is  $\sigma = 3.24$ , convergence to the optimal policy always happens and takes 5 iterations.

Secondly, we are going to compare the performance of LSPI using basis function defined on state-action graphs. The state-action graphs  $A_{SA}$ ,  $W_{SA}$ ,  $L_{SA}$  and  $\mathcal{L}_{SA}$  are constructed using trajectories of 20,000 samples and the number of basis functions is 10 which is sufficient to get convergence. The result of the experiments can be summarized as follows.

First, when using the eigenfunctions of the combinatorial Laplacian  $L_{SA}$  and the normalized Laplacian  $\mathcal{L}_{SA}$ , LSPI converges in 7 iterations to the optimal policy with probability 0.96.

Second, using shortest path Gaussian kernels on adjacency state-action graph GGKs( $A_{SA}$ ): the optimal centers are located at  $\{s_{a8}, s_{a19}, s_{a27}, s_{a41}, s_{a49}, s_{a59}, s_{a89}, s_{a79}, s_{a90}, s_{a96}\}$  and the optimal standard

deviation is  $\sigma = 7.75$ . Convergence to the optimal policy always happens and takes 6 iterations.

Third, using shortest path Gaussian kernels on weighted state-action graph  $GGKs(W_{SA})$ : the optimal centers are located at  $\{sa_6, sa_{12}, sa_{19}, sa_{41}, sa_{49}, sa_{59}, sa_{65}, sa_{79}, sa_{85}, sa_{94}\}$ , the optimal standard deviation is  $\sigma = 2.75$ . Convergence to the optimal policy always happens and takes 7 iterations.

Fourth,  $GKBs-ESP(W_{SA})$  for weighted state-action graphs: the optimal centers are located at  $\{sa_9, sa_{21}, sa_{27}, sa_{41}, sa_{50}, sa_{59}, sa_{70}, sa_{79}, sa_{90}, sa_{96}\}$ , the optimal standard deviation is  $\sigma = 7.75$ . Convergence to the optimal policy always happens and take 4 iterations. Figure 2 shows the performance of one experiment, of  $GKBs-ESP(W_{SA})$ .

The result for the 20- and 50-chain problems can be summarized as follows: The shortest path Gaussian kernel  $GKBs-ESP(W_{SA})$  gives the best result. Also, shortest path Gaussian kernels for weighted state-action graphs ( $GGKs(W_{SA})$ ) perform better than the ones for state-action graph ( $GGKs(A_{SA})$ ). Finally, with polynomial basis functions, when LSPI converges then this happens fast but often the result is a suboptimal policy. However, most of the time LSPI does not converge at all.

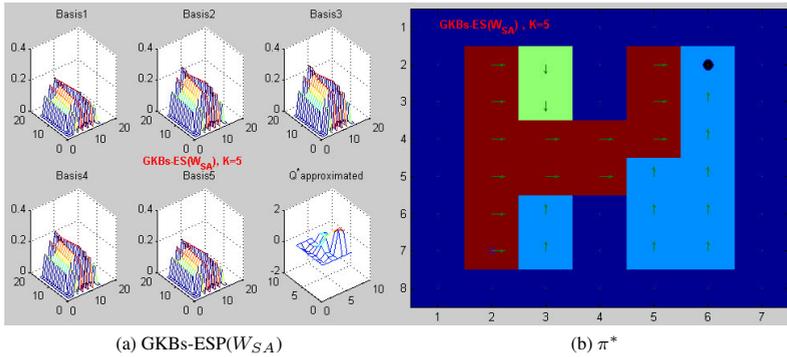


Figure 3: Performance of  $k = 5$  Gaussian kernels  $GKBs-ESP(W_{SA})$  defined on the weighted state-action graph. Subfigure (a) shows the 5 basis functions in state-action, labeled from 1 to 5, followed by the right bottom sub-subfigure which is the optimal approximated value functions,  $\hat{Q}^*$ . Subfigure (b) contains the suboptimal policy,  $\pi^*$  in each state.

For the *grid world*, fast convergence is obtained already for  $k = 5$  basis functions and the initial policy used is *Go Up* in every state of the grid. The  $\mathcal{L}_S$ -eigenbasis allow convergence to the optimal policy with probability 0.653 in 6 iterations. And, the  $\mathcal{L}_{SA}$ -eigenbasis allow convergence to the optimal policy with probability 0.71 in 4 iterations. The results for the different Gaussian kernels are as follows:

RBFs: the optimal centers and standard deviation are  $\{s_{11}, s_{13}, s_{14}, s_{21}, s_{22}\}$  and  $\sigma = 2.5$ . Convergence to the optimal policy happens with frequency 0.653 in 7 iterations.

$GGKs(A_S)$ : the optimal centers and standard deviation are  $\{s_{11}, s_{13}, s_{14}, s_{21}, s_{22}\}$  and  $\sigma = 2.5$ . Convergence to the optimal policy always happens in 5 iterations.

$GKBs-ESP(A_S)$ : the optimal centers and standard deviation are  $\{s_3, s_8, s_{16}, s_{19}, s_{22}\}$  and  $\sigma = 1.5$ . Convergence to the optimal policy happens with frequency 0.89 in 5 iterations.

$GGKs(A_{SA})$ : the optimal centers and standard deviation are  $\{s_{31}, s_{52}, s_{60}, s_{83}, s_{93}\}$  and  $\sigma = 3.75$ . Convergence to the optimal policy always happens in 7 iterations.

$GGKs(W_{SA})$ : the optimal centers and standard deviation are  $\{s_{31}, s_{52}, s_{83}, s_{85}, s_{93}\}$  and  $\sigma = 3.75$ . Convergence to the optimal policy always happens in 5 iterations.

$GKBs-ESP(W_{SA})$ : the optimal centers and standard deviation are  $\{s_{20}, s_{35}, s_{66}, s_{72}, s_{83}\}$  and  $\sigma = 4.7$ . Convergence to the optimal policy always happens in 4 iterations.

The result for the *grid world* can be summarized as follows: The best set of basis functions is again  $GKBs-ESP(W_{SA})$ , i.e. the convergence to the optimal policy is faster than for other basis. However, the construction of the basis for state-action graphs is time-consuming. For state graphs and state spaces, the shortest path Gaussian make the best basis. Finally, as the number of Gaussian kernels basis functions decreases, the standard deviation  $\sigma$  must be increased to cover the state or state-action graphs. Figure 3 shows the basis functions, the approximate optimal value function and the optimal policy in each state using  $GKBs-ESP(W_{SA})$ .

## 5 Conclusions and future work

The main conclusion is that Gaussian kernels basis functions GKBS-ESP( $W_{SA}$ ) combining the Euclidean distance and the shortest path distance on the state-action graph are working very well, i.e. LSPI converges fast. However, the construction of the state-action graph is time-consuming and the spread  $\sigma$  and center nodes must be tuned for each problem. Future work will 1) study the basis functions considered in this paper on continuous domains, e.g. the inverted pendulum, and 2) compare the accuracy of the approximated state-action value functions when either non-parametric function approximation like Gaussian processes [1] or parametric function approximation like LSPI is used.

## References

- [1] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, 2006.
- [2] F. Deutsch. *Best Approximation in Inner Product Spaces*. Springer Verlag, New-York, 2001.
- [3] F.R.K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
- [4] H. Jakab. Geodesic Distance-Based Kernel Construction For Gaussian Process Value Function Approximation. *Studia Informatica*, LVI(3):51-56, 2011.
- [5] L. Buşoniu, D. Ernst, B. De Schutter and R. Babuška. Online least-squares policy iteration for reinforcement learning control. In M. Waterfront, editor, *Proceedings of the 2010 American Control Conference*, pages 486-491, 2010.
- [6] L. Li, M.L. Littman and C.R. Mansley. Online Exploration in Least-Squares Policy Iteration. *Journal of Computer* 2:733-739, 2008.
- [7] M.G. Lagoudakis and R. Parr. Least-Squares Policy Iteration. *Journal of Machine Learning Research*, 4:1107-1149, 2003.
- [8] M.G. Lagoudakis and R. Parr. *Model-Free Least Squares Policy Iteration*. Technical Report, Computer Science Department, Duke University, Durham, North Carolina, United States, 2003.
- [9] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc., New York, USA, 1994.
- [10] M. Sugiyama, H. Hachiya, C. Towell and S. Vijayakumar. Geodesic Gaussian Kernels for Value Function Approximation. *Autonomous Robots*, 25(3):287-304, 2008.
- [11] R.R. Coifman and M. Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53-94, 2006.
- [12] R.S. Sutton and A.G. Barto. *Reinforcement learning: An introduction*. The MIT Press, 1998.
- [13] S. Mahadevan. *Representation Discovery using Harmonic Analysis*. Morgan and Claypool, 2008.
- [14] S. Mahadevan and M. Maggioni. Proto-value Functions: A Laplacian Framework for Learning Representation and Control in Markov Decision Processes. *Journal of Machine Learning Research*, 8:2169-2231, 2007.
- [15] S. Mahadevan and M. Maggioni. Value Function Approximation with Diffusion Wavelets and Laplacian Eigenfunctions. In Y. Weiss, B. Schölkopf and J. Platt, editors, *Proceedings of the Advances in Neural Information Processing Systems (NIPS 05)*, pages 843-850, 2006.
- [16] S. Osentoski and S. Mahadevan. *Action-based representation discovery in markov decision processes*. Technical Report, Computer Science Department, Massachusetts Amherst University, Amherst, Massachusetts, United States, 2009.
- [17] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

# Corpus-based Validation of a Dialogue Model for Social Support

J.M. van der Zwaan      V. Dignum      C.M. Jonker

*Delft University of Technology, P.O.Box 5010 2600 GA Delft*

## Abstract

Recent developments in affective computing show that Embodied Conversational Agents (ECAs) are increasingly capable of complex social and emotional dialogues. Our research concerns the design and evaluation of an ECA that provides social support to children that are being bullied through the Internet. Recently, we proposed a domain-independent dialogue model for verbal social support. In this paper, the model is compared to actual comforting chat conversations about bullying. Analysis of the most important and complex conversation phases shows that conversation topics predicted by the model are mostly discussed in the expected phases and that conversation patterns used to discuss topics and give (verbal) social support also correspond to the model.

## 1 Introduction

Today, children and adolescents spend a lot of time on the Internet. One of the risks they run online is to become a victim of cyberbullying. Cyberbullying is bullying through electronic communication devices. It is a complex problem that has a high impact on victims [14]. To help victims deal with their negative emotions, specialized helplines, such as Cybermentors<sup>1</sup> and Pestweb<sup>2</sup> enable them to talk to online counselors and/or peers trained to give social support. Social support or comforting refers to communicative attempts, both verbal and nonverbal, to alleviate the emotional distress of another person [5].

Early work in the field of affective computing demonstrated that virtual agents are able to reduce negative emotions in users by addressing them [10]. More recent developments show that empathic agents are increasingly capable of complex social and emotional dialogues (see for example [11, 12]). However, these dialogues are predominantly task oriented, i.e. to help users perform concrete tasks, such as finding information and learning. Generally, giving social support is unrelated to this type of tasks.

We are interested in investigating how and to what extent conversational agents can provide social support. Our research concerns the design and evaluation of an Embodied Conversational Agent (ECA) that supports cyberbullying victims. Recently, we proposed a domain-independent dialogue model for social support [19, 18]. Since it is based on psychological literature and common sense, it is unclear how well actual comforting dialogues are represented by the model. The goal of this paper is to present validation of the model in a qualitative study with real world data, i.e. actual comforting dialogues with bullying victims.

This paper is organized as follows. In section 2, we discuss related work on conversational agents that provide (social) support. In section 3, we present the domain-independent dialogue model for social support. In section 4, we specify the methodology used for the qualitative analysis and section 5 describes the results. Finally, in section 6, we present our conclusions.

## 2 Related work

Many virtual agents aimed at supporting users have been developed over the past few decades. These systems are predominantly task oriented. Another popular application of task oriented virtual agents is

---

<sup>1</sup>[www.cybermentors.org.uk](http://www.cybermentors.org.uk)

<sup>2</sup>[www.pestweb.nl](http://www.pestweb.nl)

supporting users in e-learning and tutoring systems [8, 12, 17]. Such pedagogical agents use different strategies, such as displaying active listening behavior, encouragement and praise, to motivate users and to make learning more engaging.

The ‘How was your day’ (HWYD) system developed by Cavazza et al. is an example of a non-task oriented ECA [6, 16]. The ECA allows users to talk about their day at the office and responds by asking questions to gather information or by generating appropriate affective responses to the information gathered. In addition to short sympathetic responses to the user’s input, the system may start a longer utterance that provides advice and support in a more structured fashion. These longer utterances are called comforting tirades. Comforting tirades are aimed at encouraging, comforting or warning the user.

Adam et al.’s toy that engages children with personalized dialogue is another non-task oriented dialogue system [1]. The toy’s personalization behaviors are based on the analysis of two corpora of children-adult dialogues and include strategies such as asking personal questions, recalling shared activities and taking the child’s preferences into account. Additionally, the toy employs emotional strategies including expressing empathy, encouraging the child to take active steps to remove a stressor, and providing mental disengagement [2]. These emotional strategies are based on a classification of coping strategies.

Supportive strategies of virtual agents are often based on psychological literature (e.g., [15, 13]) or expert opinions (e.g., [3]). To the best of our knowledge, these strategies have not been validated, even though some systems employing these strategies have been subjected to user experiments. We believe it is important to validate the dialogue strategies used by an ECA, especially when dealing with sensitive application domains (e.g., cyberbullying) and/or vulnerable target audiences (e.g., children). The contribution of this paper is the specification and application of a method to validate dialogue strategies with real world data.

### 3 Dialogue Model for Social Support

In this section, we present our dialogue model for comforting conversations. The model consists of multiple components, each of which is discussed in this section. The conversation is structured according to the 5-phase model, a methodology to structure counseling conversations [4]. In every conversation phase, one or more topics are discussed. Some topics have been derived from the 5-phase model while others are based on the topics that can be addressed in the course of a supportive conversation as suggested by Burleson and Goldsmith [5]. A topic is discussed in one or more dialogue sequences, where a dialogue sequence refers to a set of utterances or conversation turns in which a request for information or the pro-active sharing of a piece of information is completed by the dialogue partners. Verbal statements to communicate social support are part of these sequences. We will now discuss the different components of the model in more detail.

According to the 5-phase model, the five phases of a conversation are [4]:

1. Warm welcome: the counselor connects with the child and invites him to explain what he wants to talk about
2. Gather information: the counselor asks questions to try to establish the problem of the child
3. Determine conversation objective: the counselor and the child determine the goal of the conversation (e.g., getting tips on how to deal with bullying)
4. Work out objective: the counselor stimulates the child to come up with a solution
5. Round up: the counselor actively rounds off the conversation

During the conversation, different topics are discussed. In phase 1 the agent welcomes the user (topic hello). The topics in phase 2 (Gather information) are based on the topics that can be addressed in the course of a supportive conversation suggested by Burleson and Goldsmith [5]; these topics are: the upsetting event(s), the user’s emotional state, the personal goal that is being threatened by the upsetting event(s), and the user’s current coping strategies. In phase 3 (Determine conversation objective), the topic is the conversation objective. Even though multiple conversation objectives are possible, the model assumes the user wants to get advice on how to deal with cyberbullying. Phase 4 (Work out objective) consists again of topics suggested by Burleson and Goldsmith: future coping strategies and advice. Future coping strategies are actions the user intends to perform to cope with the problem, while advice is an action suggested by the agent for the user to perform to cope with the problem. Finally, in phase 5 (Round off) the agent says goodbye to the user (topic bye).

Support type	Description	Example
Sympathy	Express feelings of compassion or concern	How awful that you are being bullied!
Encouragement	Provide recipient with hope and confidence	I know you can do it!
Compliment	Positive assessments of the recipient and his or her abilities	Good of you to have told your parents!
Advice	Suggestions for coping with a problem	Perhaps you should tell your parents.
Teaching	Factual or technical information	You can block a contact by clicking the 'block' button

Table 1: The types of social support incorporated in the dialogue model.

Topics are discussed in one or more dialogue sequences. These sequences consist of verbal utterances expressed by either the agent or the user. During a dialogue sequence, the agent can utter speech acts to communicate social support. Phase 2 and 4, the phases in which the topics suggested by Burleson and Goldsmith are discussed, are the most appropriate to give support. Five types of social support that frequently occur in counseling conversations by chat [9] were incorporated in the model: sympathy, compliment, encouragement, advice and teaching. Table 1 lists descriptions and examples of these support types.

The model also specifies how social support is communicated in dialogue sequences for phases 2 and 4. To report sequence patterns, we use the following notation. Speech acts are denoted by  $\{Q_i, A_i, Ac_i, Sym_i, Enc_i, Com_i, Adv_i, Conf_i, Reject_i, Teach(request), Teach(step\ x)_i, SA_i\}$  which refer to question, answer, acknowledgment, sympathy, encouragement, compliment, advice, confirmation, rejection, request teaching, teach step  $x$ , and speech act (not otherwise specified) respectively;  $i \in \{c, u\}$  indicates a speech act is expressed by the counselor ( $c$ ) or the user ( $u$ ). Choices are indicated with  $|$ ,  $?$  means a speech act is uttered 0 or one times, and  $*$  indicates a speech act is repeated 0 or more times. The sequence for phase 2 and topic coping future in phase 4 is  $Q_c A_u (Ac_c | Sym_c)? (Enc_c | Com_c)*$ . The sequence for advice is:  $Adv_c Conf_u$ . Optionally, a piece of advice is followed by a list of instructions or steps (teaching). The sequence pattern for teaching is:  $Adv_c Teach(request)_c (Conf_u Teach(step\ 1)_c Conf_u \dots Teach(step\ n)_c Conf_u | Reject_u)$ . More details regarding the sequence patterns in phase 2 and 4 can be found in [18].

## 4 Methodology

To validate the dialogue model we proposed, we performed a qualitative analysis of chat conversations about bullying. These chat conversations are considered to be a 'gold standard' for our model.

### 4.1 Description of the Data

Pestweb is the Dutch center of expertise for bullying. As part of their services, they offer support to victims via telephone, chat and e-mail. The topic and setting of the chat conversations is similar to what we have in mind for the comforting dialogue agent. Three counselors gave consent to use their conversations. A total of 66 conversations were gathered over the course of one month. To protect the privacy of the children and adolescents contacting the helpline, only the counselor's side of the conversation was made available. However, the data did contain the positions of the user's utterances, so we know *when* they said something, but not *what* they said. Additionally, all utterances were anonymized; all identifying information, including the counselor's name, was replaced by generic labels. For example, proper names were replaced by <name>.

Not all 66 conversations were usable for the analysis: 17 conversations appeared to be non-serious (e.g., people trying out the helpline); 19 conversations were off-topic (e.g., conversations just before closing time, and conversations where one of the partners is experiencing technical difficulties while using the chatroom); and the conversation phases in 7 conversations were not clearly separated. The remaining 23 conversations have been analyzed. Of these, 10 were complete and 13 ended before the conversation was completed (e.g., because the user stopped responding).

## 4.2 Data Analysis

The data was coded by a single coder according to the method proposed by Chi [7]. The coder started by dividing the conversations into the phases of the 5-phase model. The second step consisted of dividing all phases into sequences of utterances. A sequence is a set of utterances or conversation turns in which a request for information or the pro-active sharing of a piece of information is completed by the dialogue partners. In step 3, all sequences were assigned a topic. Coding started from the topics proposed by the model and codes were added for topics not covered by the model.

A codes was added for ‘comforting tirades’ (cf. [6, 16]) which are longer utterances to provide support that can be characterized as advice nor teaching, but, as for advice and teaching, the whole sequence is dedicated to giving support. An example of such a sequence is ‘*You know bullying is really complicated, let me explain something to you. Bullies want to stand out. Sometimes by bossing people around. Or by making fun of others. They think others look up to them for what they do. But usually that’s not true (...)*’ (conversation 39). Additionally, new codes were invented for conversation and chat management. Conversation management includes feedback requests, summaries and other techniques, whereas chat management refers to sequences dealing with technical difficulties during the chat. Since the topics proposed by the model are not discussed in these kinds of sequences, they have been separated from the sequences in which topics proposed by the model were discussed. Finally, the code off-topic was added. Sequences coded as off-topic deal with topics that are outside the scope of a chat conversation about bullying. For example, the counselors asked some of the children to fill out a questionnaire about their experiences during the conversation. These sequences have been coded as off-topic.

Step 4 of the coding process consisted of marking social support types in the sequences of phase 2 and 4. Finally, sequence patterns occurring in phase 2 and 4 of the conversations were extracted. Again, we started with the sequence patterns as proposed by the dialogue model. The proposed patterns were changed to better fit the data; in the QA-pattern  $Ac_c$  was made optional, because in the data, the counselor does not always respond to an answer provided by the user. Additionally, the user does not always respond to a piece of advice, so the user confirmation in the advice sequence was made optional as well. Also, codes for new patterns were added. In order to describe newly found utterances, we introduced  $\{Adv(introduction)_i, Adv(options)_i\}$  to define parts of a new type of advice. More details about the sequence patterns in the data can be found in section 5.2.

While coding, we made some assumptions about the data. Because the data consists of utterances and the proposed sequence patterns consist of speech acts, we assume an utterance consists of one or more speech acts and a speech act can consist of one or more utterances. The utterances of the user are unavailable in the data, but the user turns are available. We assume a user turn can consist of multiple utterances and/or multiple speech acts. The contents of a user turn are determined based on the response of the counselor. We assume that a user utterance is relevant only if the counselor explicitly responds to it. While a few sequences were difficult to understand based on the counselor utterances alone, in general, the conversations were easy to follow, because the counselors frequently request feedback from their conversation partners and summarize their input.

## 5 Results

In this section we describe the results of the data analysis and assess the match between the data and the model.

### 5.1 Phases and Topics

In total 637 sequences were found in the data; 534 of these (83.8%) was coded with one of the topics from the model. Table 2 shows the number of conversations in which a topic is discussed for each of the 5 conversation phases. As shown by the grey cells, topics occur generally in the phases where they belong according to the model. Only in phase 4 topics from other phases occur regularly.

The topic hello (H) occurs exclusively in phase 1 of all conversations. The topics that belong in phase 2 are: event (E), emotional state (ES), personal goal (PG) and coping current (CC). The event is discussed in phase 2 of all conversations, in some conversations in phase 4, and once in phase 3. The data shows that the emotional state (ES) is discussed in phase 2 of about half of the conversations. Personal goal (PG) is hardly discussed. There are no occurrences in phase 2 and only in one conversation in phase

	H	E	ES	PG	CC	CO	CF	A	B	Total
Phase 1	23									23
Phase 2		23	10	0	17					23
Phase 3		1				13				13
Phase 4		5		1	4	1	12	11		14
Phase 5							1	1	16	16

Table 2: Occurrence of topics in the different phases. Grey cells indicate that, according to the model, the topic occurs in this phase. (H: hello, E: event, ES: emotional state, PG: personal goal, CC: coping current, CO: conversation objective, CF: coping future, A: advice, B: bye, Total: Total number of conversations containing this phase (at least partial))

4. Apparently, this is not a common topic. As expected, coping current (CC) occurs in phase 2 of most conversations, but this topic also occurs in phase 4. The conversation objective (CO) is discussed almost exclusively in phase 3. Coping future (CF) is discussed almost exclusively in phase 4 and it occurs in most conversations, as is advice (A). Finally, the topic bye (B) is discussed in all conversations that include a phase 5.

## 5.2 Sequence Patterns in phase 2 and 4

The results so far gave a general impression of the sequences that occur in the conversations. Topics are discussed by one or more sequences. The proposed model specifies dialogue patterns for sequences in phase 2 and 4. In this section we explore to what extent these patterns can be found in the data. A complicating factor is that the sequence patterns were specified in speech acts and the data consists of utterances. We assume a speech act consists of one or more utterances and an utterance consists of one or more speech acts. Additionally, we only include user utterances in the patterns if the counselor explicitly responds to a user utterance (we assume that the counselor explicitly responds to user utterances that contain relevant information).

Table 3 contains all sequence patterns we found in phase 2 and 4. Included in the analysis are all sequences with topics event, emotional state, personal goal, coping current, coping future, advice, and social support other. The sequence patterns are specified in the second column of table 3. The third column contains examples from the data. To indicate the user utterances are missing from the data, they have been replaced by black squares; the position of the user utterances does come from the corpus.

All 23 conversations in the corpus contain a (partial) phase 2. In this data, we found 7 different sequence patterns; the one specified by the model (QA 1) and 6 new ones (QA 2, RtU 1-3, Other 1 and 2). Table 4 shows the frequency of each of these patterns in phase 2. Clearly, pattern QA 1 is the most important pattern; it occurs in all 23 conversations and accounts for 94.2% of the sequences.

The corpus contains 14 conversations with a (partial) phase 4. In this data, we found 6 different sequence patterns; the three specified by the model (QA 1, Advice 1 and Teaching), a new one (Advice 2) and 2 patterns that were also found in phase 2 (RtU 1 and Other 1). Table 5 shows the frequencies of these patterns in phase 4. Patterns QA 1, RtU 1 and Other 1 are 'general' conversation patterns used to discuss coping future and other topics in phase 4, whereas Advice 1, Advice 2 and Teaching are used to give advice. Pattern QA 1 accounts for 66.2% of the general conversation sequences and patterns Advice 1 and Teaching account for 66.7% of the advice giving sequences.

## 5.3 Match between the Data and the Model

In general, the model predicts the occurrence of the topics in the phases very well. Only in phase 4 some topics not expected in this phase are discussed. Additionally, the topic Personal goal seems to be irrelevant as it only occurs once in the 23 conversations analyzed. Over 80% of the sequences in the data could be assigned one of the topics from the model. Sequences that could not be assigned a topic from the model are conversation or chat management techniques (6.8%), off-topic (5.5%) or a new kind of supportive utterance that can be characterized as a 'comforting tirade' (3.9%).

Additionally, we analyzed the patterns in conversation sequences for phases 2 and 4. After relaxing two of the three patterns specified by the model, we found that a majority of the sequences followed the (relaxed) patterns (94.2% for phase 2 and over 66% for phase 4). In total, 7 additional sequence patterns were found.



So, the conversations show a lot of regularity and again we can conclude there is a good match between the data and the model. However, the analysis shows that the dialogue model misses a pattern to respond to user input without asking a question first. While only 5.56% of the sequences in phase 2 and 4 was assigned pattern RtU 1, we believe it is important the comforting ECA is able to respond to information the user introduces pro-actively. Therefore, this pattern will be added to the model.

## 6 Conclusion

In this paper, we compared a dialogue model for social support based on psychological literature and common sense to the counselor side of 23 real comforting chat conversations. The results show great similarities between the data and the model on the topics discussed, the phases in which these topics are discussed, and the sequence patterns used to discuss topics and give (verbal) social support.

To further confirm the validity of the dialogue model, more research is needed. First of all, to improve the validity of the analysis, more coders should be involved in analyzing the data. Second, while 23 conversations is a substantial amount, the generalizability of the results could be increased by analyzing more conversations. And finally, to investigate the extend to which the conversation model is domain independent, social support conversations from other domains should be included in the analysis as well.

## 7 Acknowledgements

This work is funded by NWO under the Responsible Innovation (RI) program via the project ‘Empowering and Protecting Children and Adolescents Against Cyberbullying’.

## References

- [1] C. Adam, L. Cavedon, and L. Padgham. “Hello Emily, how are you today?”: personalised dialogue in a toy to engage children. In *Proceedings of the 2010 Workshop on Companionable Dialogue Systems*, CDS ’10, pages 19–24, 2010.
- [2] C. Adam and P. Ye. Reasoning about emotions in an engaging interactive toy (extended abstract). In *Proceedings of the 8th International Conference on Autonomous agents and Multiagent Systems (AAMAS 2009)*, pages 31–32, 2009.
- [3] I. Arroyo, K. Muldner, W. Bursleson, B. Woolf, and D. Cooper. Designing Affective Support to Foster Learning, Motivation and Attribution. In *AIED 2009: 14th Int. Conf. on Artificial Intelligence in Education Workshops Proceedings*, 2009.
- [4] A. de Beyn. *In gesprek met kinderen: de methodiek van de kindertelefoon*. SWP, 2003.
- [5] B.R. Bursleson and D.J. Goldsmith. *Handbook of Communication and Emotion: Research, Theory, Applications, and Contexts*, chapter How the Comforting Process Works: Alleviating Emotional Distress through Conversationally Induced Reappraisals, pages 245–280. Academic Press, 1998.
- [6] M. Cavazza, C. Smith, D. Charlton, N. Crook, J. Boye, S. Pulman, K. Moilanen, D. Pizzi, R. de la Camara, and M. Turunen. Persuasive dialogue based on a narrative theory: An eca implementation. In T. Ploug, P. Hasle, and H. Oinas-Kukkonen, editors, *Persuasive Technology*, volume 6137 of *LNCS*, pages 250–261. Springer, 2010.
- [7] M. Chi. Quantifying qualitative analyses of verbal data: a practical guide. *Journal of the Learning Sciences*, 6:217–315, 1997.
- [8] S. D’Mello, B. Lehman, J. Sullins, R. Daigle, R. Combs, K. Vogt, L. Perkins, and A. Graesser. *Intelligent Tutoring Systems*, volume 6094 of *LNCS*, chapter A Time for Emoting: When Affect-Sensitivity Is and Isn’t Effective at Promoting Deep Learning, pages 245–254. Springer, 2010.
- [9] R. Fukkink. Peer counseling in an online chat service: A content analysis of social support. *Cyberpsychology, Behavior, and Social Networking*, 14(4):247–251, 2011.

- [10] K. Hone. Empathic agents to reduce user frustration: The effects of varying agent characteristics. *Interact. Comput.*, 18(2):227–245, 2006.
- [11] S. Kopp, L. Gesellensetter, N.C. Krämer, and I. Wachsmuth. *Intelligent Virtual Agents*, chapter A Conversational Agent as Museum Guide – Design and Evaluation of a Real-World Application, pages 329–343. 2005.
- [12] T.-Y. Lee, C.-W. Chang, and G.-D. Chen. Building an interactive caring agent for students in computer-based learning environments. In *Proceedings of the 7th IEEE Int. Conf. on Advanced Learning Technologies, ICALT 2007*, pages 300–304, 2007.
- [13] C.L. Lisetti and E. Wagner. Mental health promotion with animated characters: Exploring issues and potential. In *AAAI Spring Symposium: Emotion, Personality, and Social Behavior*, pages 72–79, 2008.
- [14] S. Livingstone, L. Haddon, A. Görzig, and K. Ólafsson. Risks and safety on the internet: the perspective of European children: full findings. <http://eprints.lse.ac.uk/33731/>, 2011.
- [15] R. Looije, M.A. Neerincx, and F. Cnossen. Persuasive robotic assistant for health self-management of older adults: Design and evaluation of social behaviors. *International Journal of Human-Computer Studies*, 68(6):386–397, 2010.
- [16] C. Smith, N. Crook, J. Boye, D. Charlton, S. Dobnik, D. Pizzi, M. Cavazza, S. Pulman, R. de la Camara, and M. Turunen. Interaction strategies for an affective conversational agent. In J. Allbeck, N. Badler, T. Bickmore, C. Pelachaud, and A. Safonova, editors, *Intelligent Virtual Agents*, volume 6356 of *LNCS*, pages 301–314. Springer, 2010.
- [17] K. Zakharov, A. Mitrovic, and L. Johnston. Towards emotionally-intelligent pedagogical agents. In *Proceedings of the 9th international conference on Intelligent Tutoring Systems, ITS '08*, pages 19–28. Springer-Verlag, 2008.
- [18] J.M. van der Zwaan, V. Dignum, and C.M. Jonker. A bdi dialogue agent for social support: Specification and evaluation method. In *Proceedings of the 3rd Workshop on Emotional and Empathic Agents @ AAMAS 2012*, 2012.
- [19] J.M. van der Zwaan, V. Dignum, and C.M. Jonker. A conversation model enabling intelligent agents to give emotional support. In *Proceedings of the 25th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE 2012)*, Dalian, China, 2012.



# **Extended Abstracts**

**BNAIC 2012**



# Mapping Product Taxonomies in E-commerce

Steven Aanen      Lennart Nederstigt      Damir Vandić      Flavius Fräsincar

*Econometric Institute, Erasmus University Rotterdam  
P.O. Box 1738, 3000 DR Rotterdam, the Netherlands*

The full version of this paper, entitled ‘SCHEMA - AN ALGORITHM FOR AUTOMATED PRODUCT TAXONOMY MAPPING IN E-COMMERCE’ appeared in: Extended Semantic Web Conference (ESWC 2012), volume 7295 of Lecture Notes in Computer Science, pages 300-314. Springer, 2012.

## Abstract

In this paper we propose SCHEMA, an algorithm that automatically maps heterogeneous product taxonomies in the domain of e-commerce. SCHEMA employs a custom word sense disambiguation technique, based on the Lesk algorithm, in combination with the semantic lexicon WordNet. For finding candidate target categories and determining the path-similarity we propose a semantic category matching algorithm that takes into account the disambiguation process of a category. The mapping quality score is calculated using the Damerau-Levenshtein distance and a node-dissimilarity penalty. The performance of SCHEMA was tested on three real-life datasets and compared to PROMPT and the algorithm proposed by Park & Kim. The comparison shows that SCHEMA improves considerably recall and  $F_1$ -score, while maintaining similar precision.

## 1 Introduction

In recent years the Web has increased dramatically in both size and range, playing an increasingly important role in our society and world economy. For instance, the estimated revenue for e-commerce in the USA grew from \$7.4 billion in 2000 to \$34.7 billion in 2007 [2]. As a consequence, the aggregation of product data is becoming increasingly important. A common problem encountered in this task is the mapping of product taxonomies from different Web stores to an existing product taxonomy. By matching the product taxonomies from different Web stores, it becomes easier to compare products.

As a solution we propose the *Semantic Category Hierarchy for E-commerce Mapping Algorithm*, also called SCHEMA. This algorithm can be used to map heterogeneous product taxonomies from multiple sources to each other. The algorithm employs a word sense disambiguation technique that is based on the *Lesk algorithm* [3] to find synonyms of the correct sense for the source category name. Furthermore, it uses lexical similarity measures, such as the *Levenshtein distance*, along with structural information, to determine the best candidate category to map to. In order to evaluate SCHEMA, its performance is compared on recall and precision with PROMPT [4] and the algorithm proposed by Park & Kim [5].

## 2 SCHEMA

The SCHEMA algorithm takes as input a source taxonomy and a target taxonomy. For each category in the source taxonomy the algorithm produces a mapping to the target taxonomy. The algorithm can also provide a ‘blank’ mapping, in this case the best match in the target taxonomy did not exceed a certain quality threshold. SCHEMA executes the following three main steps for each category in the source taxonomy, which we will call the ‘source category’ from now on. The first step is to disambiguate the source category, which results in obtaining a set of synonyms of the correct sense. The second step involves selecting candidate categories from the target taxonomy using the set of synonyms obtained in the previous step. In the third step a comparison is performed with the source category to select the best-fitting candidate target category.

In the first step the algorithm first applies a splitting procedure on the parent of the source category, the source category itself, and the the source category's children. For example, the category "Music & Video" would be divided in the split terms 'Music' and 'Video'. The union of all split terms is considered as the context for the word sense disambiguation algorithm, which is based on the Lesk algorithm. The result of this step is the extended split term set, which is a set of synonym sets where each synonym set corresponds to a split term in the source category. In this step, as well as in subsequent steps, the Levenshtein distance is used to compare single terms.

The second step uses the extended split term set to select candidate target categories. This is done by also taking into account composite categories, e.g., 'Music & Video'. The issue with composite categories is that we do not want to map 'Music & Video' to 'Music', whereas mapping 'Music' to 'Music & Video' is fine. The way SCHEMA deals with this issue is that it uses each synonym set in the extended split term set to check for a match between two categories. The match checking is performed by the proposed category matching algorithm, which is based on the *longest common substring similarity*. A target category becomes a candidate if all the synonym sets of the source category are matching that target category.

The third step involves comparing candidate category paths to the source category path. For this purpose, we use the Damerau-Levenshtein distance [1]. This procedure computes a similarity between the source category and each candidate target category, taking into account the structure of the category paths. The mapping is selected by taking the candidate with the highest similarity. A threshold is used to avoid mapping to an unsuitable target taxonomy.

### 3 Evaluation and Conclusion

Three product taxonomies from real-life datasets were used for the evaluation. We used Amazon (2,500 categories), Overstock.com (1,000), and the Open Directory Project (44,000 categories). Using these three datasets, six different combinations of source and target taxonomies were performed. Using a sample of 500 for each data set, we manually mapped  $6 \times 500 = 3000$  categories.

For the  $F_1$ -score, the evaluation shows that PROMPT has 20.75%, the Park & Kim algorithm 32.52%, and SCHEMA 55.10%. For the recall, these values are 16.69% for PROMPT, 25.19% for Park & Kim, and 80.73% for SCHEMA. For the precision, these values are 28.93% for PROMPT, 47.77% for Park & Kim, and 42.21% for SCHEMA. The results of our evaluation show that SCHEMA performs better than PROMPT and the algorithm of Park & Kim, on both recall and  $F_1$ -score, while maintaining a similar precision. This can be attributed to the ability of SCHEMA to cope with lexical variations in category names, as well as the ability to properly deal with composite categories.

In conclusion, the main objective for developing the SCHEMA algorithm was to facilitate the aggregation of product information from different Web sources by providing a product taxonomy mapping algorithm. In order to improve the recall, our algorithm employs word sense disambiguation and addresses the recurring issue of composite product categories. The performance of our algorithm was tested on three real-life datasets and compared with the performance of PROMPT and the algorithm of Park & Kim. This evaluation shows that SCHEMA achieves a considerably higher average recall than the other algorithms, while maintaining a similar precision.

### References

- [1] Fred J. Damerau. A Technique for Computer Detection and Correction of Spelling Errors. *Communications of the ACM*, 7(3):171–176, 1964.
- [2] John B. Horrigan. Online Shopping. *Pew Internet & American Life Project Report*, 36, 2008.
- [3] Michael Lesk. Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *5th Annual International Conference on Systems Documentation (SIGDOC 1986)*, pages 24–26. ACM, 1986.
- [4] Natalya F. Noy and Mark A. Musen. The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
- [5] Sangun Park and Wooju Kim. Ontology Mapping between Heterogeneous Product Taxonomies in an Electronic Commerce Environment. *International Journal of Electronic Commerce*, 12(2):69–87, 2007.

# Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies\*

## *Extended Abstract*

Tim Baarslag

Koen Hindriks

Mark Hendriks

Alex Dirkzwager

Catholijn M. Jonker

*Interactive Intelligence Group, Delft University of Technology,  
Mekelweg 4, Delft, The Netherlands*

*{T.Baarslag, K.V.Hindriks, M.J.C.Hendriks, C.M.Jonker}@tudelft.nl*

## 1 Introduction

In the last decade, many different negotiation strategies have been introduced in the search for an effective, generic automated negotiator. There is now a large body of negotiation strategies available, and with the emergence of the International Automated Negotiating Agents Competition (ANAC) [1], new strategies are generated on a yearly basis. While methods exist to determine the best negotiation agent given a set of agents [1], we still do not know which type of agent is most effective in general, and especially why. As it is impossible to exhaustively search the large space of negotiation strategies, there is a need for a systematic way of searching this space for effective candidates.

Many of the sophisticated agent strategies that currently exist are comprised of a fixed set of modules. Generally, a distinction is made between three different modules: one module that decides whether the opponent's bid is acceptable; one that decides which set of bids could be proposed next; and finally, one that tries to guess the opponent's preferences and takes this into account when selecting an offer to send out.

The negotiation strategy is a result of the complex interaction between these components, of which the individual performance may vary significantly. For instance, an agent may contain a module that predicts the opponent's preferences very well, but the agent may still perform badly utility-wise because it concedes far too quickly. This means that overall performance measures, such as average utility obtained in a tournament, make it hard to pinpoint which components of an agent work well. To date no efficient method exists to identify to which of the components the success of a negotiating agent can be attributed. Finding such a method would allow to develop better negotiation strategies, resulting in better agreements; the idea being that well-performing components together will constitute a well-performing agent.

## 2 The BOA Agent Framework

Based on a survey of literature and the implementations of currently existing negotiation agents, we propose to analyze three components of the agent design separately. We show that most of the currently existing negotiating agents can be fitted into the so-called *BOA framework* by putting together the following three main components in a particular way:

1. **Bidding strategy.** A bidding strategy is a mapping which maps a negotiation trace to a bid. It can interact with the opponent model by consulting with it, passing one or multiple bids and see how they compare within the opponent's utility space.

---

\*The full version of this paper appears in: Tim Baarslag, Koen Hindriks, and Catholijn M. Jonker. Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies. In Proceedings of the Fifth International Workshop on Agent-based Complex Automated Negotiations (ACAN'12). Valencia, Spain, June 2012.

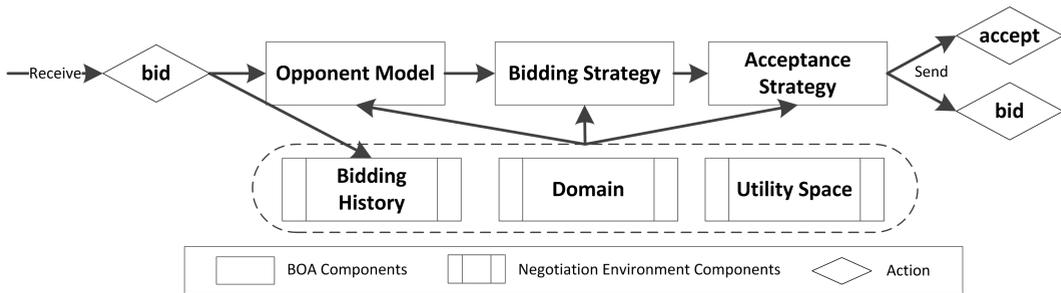


Figure 1: The BOA framework negotiation flow.

2. **Opponent model.** An opponent model is a learning technique that constructs a model of the opponent's negotiation profile.
3. **Acceptance strategy.** The acceptance strategy determines whether the bid that the opponent has presented is acceptable.

The components interact in the following way (the full process is visualized in Figure 1). When receiving an opponent bid, the BOA agent first updates the *bidding history* and *opponent model* to make sure the most up-to-date data is used. Given the opponent bid, the *bidding strategy* determines the counter offer by first generating a set of bids with a similar preference for the agent. The *bidding strategy* uses the *opponent model* (if present) to select a bid from this set by taking the opponent's utility into account. Finally, the *acceptance strategy* decides whether the opponent's action should be accepted; if not, the bid generated by the bidding strategy is offered instead.

The advantages of fitting agents into the BOA framework are threefold: first, it allows to study the behavior and performance of individual components; second, it allows to systematically explore the space of possible negotiation strategies; third, the identification of unique interacting components simplifies the creation of new negotiation strategies.

### 3 Results and Conclusion

This paper introduces a framework that distinguishes the bidding strategy, the opponent model, and the acceptance strategy in automated negotiation strategies and recombines these components to systematically explore the space of automated negotiation strategies. The main idea behind the BOA framework is that we can identify several components in a negotiating agent, all of which can be optimized individually.

Our scientific contribution is twofold: first, we show that existing state-of-the-art agents (including the agents from ANAC 2010 [1] and 2011) are compatible with this architecture by re-implementing them in the new framework, while demonstrating that the original agents and their decoupled versions have identical behavior and similar performance; secondly, as an application of our architecture, we systematically explore the space of possible strategies by recombining different strategy components, resulting in negotiation strategies that improve upon the current state-of-the-art in automated negotiation.

### References

- [1] Tim Baarslag, Koen Hindriks, Catholijn M. Jonker, Sarit Kraus, and Raz Lin. The first automated negotiating agents competition (ANAC 2010). In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence*, pages 113–135, Berlin, Heidelberg, 2012. Springer-Verlag.

# Nested Monte-Carlo Tree Search for Online Planning in Large MDPs<sup>1</sup>

Hendrik Baier

Mark H. M. Winands

*Games and AI Group, Department of Knowledge Engineering, Maastricht University,  
The Netherlands*

*Monte-Carlo Tree Search* (MCTS) [3, 5] is a promising choice for online planning in large MDPs. It is a best-first, sample-based search algorithm in which every state in the search tree is evaluated by the average outcome of Monte-Carlo rollouts from that state. Since MCTS is based on sampling, it does not require a transition function in explicit form, but only a generative model of the domain. Because it grows a highly selective search tree guided by its samples, it can handle search spaces with large branching factors. By using Monte-Carlo rollouts, MCTS can take long-term rewards into account even with distant horizons. Combined with multi-armed bandit algorithms to trade off exploration and exploitation, MCTS has been shown to guarantee asymptotic convergence to the optimal policy [5], while providing approximations when stopped at any time.

For the consistency of MCTS, i.e. for the convergence to the optimal policy, uniformly random rollouts beyond the tree are sufficient. However, heuristically informed rollout strategies typically speed up convergence [4]. In this paper, we propose *Nested Monte-Carlo Tree Search* (NMCTS), using the results of lower-level searches recursively to provide rollout policies for searches on higher levels.

So far, no nested search algorithm has made use of the selectivity and exploration-exploitation control that MCTS provides. In the context of MCTS, nested search has so far only been performed offline to provide opening databases for the underlying online game playing agent. The different levels of search therefore used different tree search algorithms adapted to their respective purpose, and nested and regular MCTS have not been compared on the same task. In this extended abstract, we propose Nested Monte-Carlo Tree Search (NMCTS) as a general online planning algorithm for MDPs.

We define a level-0 *Nested Monte-Carlo Tree Search* (NMCTS) as a single rollout with the base rollout policy—either uniformly random, or guided by a simple heuristic. A level-1 NMCTS search corresponds to MCTS, employing level-0 searches as state evaluations. A level- $n$  NMCTS search for  $n \geq 2$  recursively utilizes the results of level- $(n - 1)$  searches as evaluation returns.

As the selection, expansion and backpropagation steps of MCTS are preserved in NMCTS, many successful techniques from MCTS research such as the UCB1-TUNED selection policy can be applied in NMCTS as well. Parameters can be tuned for each level of search independently.

In some domains, it is effective not to spend the entire search time on the initial position of a problem, but to distribute it over all actions in the episode (or the first  $z$  actions). Search and execution are thus interleaved. We call this technique *action-by-action search* as opposed to *global search*, and it is optionally applicable at all levels of NMCTS. In case NMCTS is used with action-by-action search, a decision has to be made which action to choose and execute at each step of the search. Two possible options are a) choosing the most-sampled action—as traditionally done in MCTS—, or b) choosing the next action in the overall best solution found so far. Setting NMCTS to action-by-action search, using only one rollout per legal action in each action search, and then choosing the next action of the best known solution leads to NMCS [2] as a special case of NMCTS. This special case does not provide for an exploration-exploitation tradeoff, nor does it build a tree going deeper than the number of nesting levels used, but it allows relatively deep nesting due to the low number of rollouts per search level.

---

<sup>1</sup>This work is funded by the Netherlands Organisation for Scientific Research (NWO) in the framework of the project Go4Nature, grant number 612.000.938.

The full version of this paper is published in: *20th European Conference on Artificial Intelligence (ECAI 2012)*, pp. 109-114, 2012.

We tested Nested Monte-Carlo Tree Search on three different deterministic, fully observable MDPs: The puzzles named “SameGame”, “Clickomania” and “Bubble Breaker”. These domains have identical transition functions, but different reward functions, resulting in different distributions of high-quality solutions. The decision problem associated with these optimization problems is NP-complete [1].

We compared regular MCTS and level-2 NMCTS in all three domains, using a random rollout policy. For SameGame, we also employed a state-of-the-art informed rollout policy, consisting of the TabuColor-RandomPolicy [6] (setting a “tabu color” at the start of each rollout that is not chosen as long as groups of other colors are available) in combination with a multi-armed bandit learning the best-performing tabu color for the position at hand (based on UCB1-TUNED).

As it has been shown for SameGame that restarting several short MCTS runs on the same problem can lead to better performance than a single, long run [6], we tested several numbers of randomized restarts for MCTS and tuned the selection policy for each of them. The same settings were then used for NMCTS, with the number of nested level-1 NMCTS searches equivalent to the number of restarts for multi-start MCTS.

Results show that in Bubble Breaker and SameGame—in the latter using both random and informed rollouts—level-2 NMCTS significantly outperformed multi-start MCTS in all experimental conditions. The best results in SameGame were achieved building a level-2 tree out of 36,480 level-1 searches of 250 ms each, with informed base-level rollouts. In comparison to the best performance of multi-start MCTS, achieved with 2280 restarts of 4-second searches, the use of a nested tree increased the average best solution per position from 3395.9 to 3465.96. As a comparison, a doubling of the search time to 4560 restarts only resulted in a performance increase to 3431.0.

In Clickomania, level-2 NMCTS also achieved the highest score. While the results of multi-start MCTS for different numbers of restarts suggest that a single, global MCTS search could perform relatively well in Clickomania, memory limitations reduced the effectivity of this approach. NMCTS however is able to constantly reuse tree nodes of lower-level searches, and therefore suffers less from this problem. We observed that the best-performing NMCTS setting tested used less than 15% of memory of what a single, global MCTS search would have required for optimal performance.

In further experiments, we compared level-2 NMCTS to level-3 NMCS. Here, NMCTS used action-by-action search on level 2, and advanced from action to action by choosing the next action of the best solution found so far. NMCS was not able to complete a level-3 search in the given time; consequently, the best solutions found after the given computation time had elapsed were used for the comparisons. NMCTS outperformed NMCS in SameGame with random playouts, SameGame with informed playouts and Clickomania. For Bubble Breaker, manual tuning has not revealed parameter settings superior to NMCS yet. Automatic parameter tuning is in preparation.

In conclusion, empirical results in the test domains of SameGame, Bubble Breaker and Clickomania show that NMCTS significantly outperforms regular MCTS. Experiments in SameGame and Clickomania suggest performance superior to NMCS. Since both MCTS and NMCS represent specific parameter settings of NMCTS, correct tuning of NMCTS has to lead to greater or equal success in any MDP domain.

## References

- [1] T. C. Biedl, E. D. Demaine, M. L. Demaine, R. Fleischer, L. Jacobsen, and J. I. Munro. The Complexity of Clickomania. In R. J. Nowakowski, editor, *More Games of No Chance, Proceedings of the MSRI Workshop on Combinatorial Games*, pages 389–404, 2002.
- [2] T. Cazenave. Nested Monte-Carlo Search. In C. Boutilier, editor, *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 456–461, 2009.
- [3] R. Coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In H. J. van den Herik, P. Ciancarini, and H. H. L. M. Donkers, editors, *5th International Conference on Computers and Games (CG 2006). Revised Papers*, volume 4630 of *Lecture Notes in Computer Science*, pages 72–83. Springer, 2007.
- [4] S. Gelly, Y. Wang, R. Munos, and O. Teytaud. Modification of UCT with Patterns in Monte-Carlo Go. Technical report, HAL - CCSD - CNRS, France, 2006.
- [5] L. Kocsis and C. Szepesvári. Bandit Based Monte-Carlo Planning. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *17th European Conference on Machine Learning, ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer, 2006.
- [6] M. P. D. Schadd, M. H. M. Winands, H. J. van den Herik, G. M. J.-B. Chaslot, and J. W. H. M. Uiterwijk. Single-Player Monte-Carlo Tree Search. In H. J. van den Herik, X. Xu, Z. Ma, and M. H. M. Winands, editors, *Proceedings of the 6th International Conference on Computers and Games*, pages 1–12, 2008.

# Providing Feedback for Common Problems in Learning by Conceptual Modeling using Expectation-Driven Consistency Maintenance

Wouter Beek <sup>a</sup>

Bert Bredeweg <sup>a</sup>

<sup>a</sup> *Informatics Institute, University of Amsterdam*

The full paper version of this paper is published in the Proceedings of the 26th International Workshop on Qualitative Reasoning (QRW 2012).

## 1 Introduction

Learning by conceptual modeling is a strong paradigm for learning, allowing students to express and externalize their thinking. Environments exist that allow conceptual models to be constructed and simulated [3, 1]. These tools employ a qualitative vocabulary for users to construct their explanations of phenomena, notably about systems and how they behave. Although these tools are promising in the learning effect that they bring about, progress gets hampered when learners want, but are unable to adjust their model so that the simulation results align with their expectations.

## 2 Cognitive Model-Based Diagnosis

In technical diagnosis [5], faulty components that cause a device to behave inappropriately are singled out. Technical diagnosis was adjusted for cognitive diagnosis [2], the process of inferring a person's cognitive state from his or her performance. In particular, cognitive diagnosis can be used to find the faulty inference steps in a learner's reasoning. In technical diagnosis a norm model drives the diagnostic algorithm. Similarly, in cognitive diagnosis learners interact with an existing model (created by experts or teachers) and faulty answers are diagnosed using this model as the norm.

However, interacting with an existing (norm) model does not line up well with contemporary theories on 'active learning' originating from constructivist perspectives on learning. In order to facilitate active learning, we developed an approach that takes the discrepancies between the actual and the learner-expected simulation results of a model, and then identifies those aspects of the learner-built model that are accountable for those differences. In this way, the learner's knowledge construction endeavor is supported by maintaining consistency between the expression created by the learner and the expectation s/he holds regarding the inferences that can be made on behalf of that expression.

Conceptual models present extra complexity for model-based cognitive diagnosis, since they use weak constraints (qualitative instead of quantitative), hampering reasoning with component behavior rules. Another challenge is to warrant tractability. Aggregations made by experts in a norm model are used to put the focus on a particular subset of all possible system behaviors, which most likely does not include the modeler's expected behavior. An aggregation algorithm is therefore developed that takes the modeler's expected behavior as the main driver. Since a learner may express any expectation, this tuning of the representation must occur 'online' and be able to work for an extensive number of cases.

### 3 Approach

The output from simulations run inside the DynaLearn ILE [1] is transformed into a representation that is useful for diagnosis, the Component Connection Model (CCM). Components are explicit representations of instances of deduction steps the learner must grasp in order to understand the simulation results. Deduction steps are applied to the expressions that are asserted in the connections between the ports of the components. The components are added to the CCM based on an extendable library of component definitions, specifying the number of ports, the port types, and the behavioral description of the modeled deduction step.

Based on the generated CCM representation, the learner is presented with a graphical user interface in which expectations regarding the simulation results can be formulated. After expectations have been added to the CCM, component aggregation is performed relative to these expectations. There are two types of aggregations: competitive and hierarchical. Competitive components subsume lower-level components that need to be taken into account together in order to explain their behavior (e.g., competing processes).

Hierarchical aggregation subsumes components whose behavior is understood individually, but which can also be calculated when taken together. Hierarchical aggregation reduces the within-state complexity of the CCM in terms of the number of component assumptions that have to be made in diagnosis. In addition, it allows for a better way to steer the diagnostic interaction, since it allows discrepancies to be noted at a higher level of abstraction. Aggregate components that are part of conflict sets are unpacked so that diagnosis can proceed at a lower level of the CCM representation.

Typically, a single diagnosis iteration does not narrow the discrepancies down to the lowest level of detail. In such cases, more information from the learner is needed. At the end of each diagnosis iteration the possibilities for gaining additional knowledge are determined. The connections in the CCM are possible locations for probing the learner. Since probing means asking a learner questions, we want to perform as little probes as possible to keep the interaction smooth. This is achieved by ordering these locations according to relevance (the degree in which their knowledge would allow the diagnosis algorithm to exclude candidate hypotheses).

The diagnosis ends when there is only one candidate, i.e., the best possible guess of what causes the discrepancy between expectation and simulation results given the expectations and probe answers. It is not always necessary to run a diagnosis until it ends. As soon as all candidates share a component, this can already be communicated as a potential cause of inconsistency. The learner can inspect intermediary outcomes and is in control as to how long diagnosis takes.

### 4 Conclusion

The expectation-based diagnostic approach is able to solve inconsistencies in conceptual models for often occurring problem categories that were identified based on classroom studies. The algorithms have been fully implemented and integrated into the DynaLearn Integrated Learning Environment (ILE). Our current implementation is able to give useful feedback on difficult modeling problems for which no alternative form of automated feedback is given in comparable qualitative learning environments [4, 3]. Based on an expert review of the performance on models from the identified problem categories, the current approach seems fruitful for considering other types of conceptual modeling inconsistencies in the future.

### References

- [1] B. Bredeweg, F. Linnebank, A. Bouwer, and J. Liem. Garp3: Workbench for qualitative modelling and simulation. *Ecological Informatics*, 4(5-6):263–281, 2009.
- [2] K. de Koning, B. Bredeweg, J. Breuker, and B. Wielinga. Model-based reasoning about learner behaviour. *Artificial Intelligence*, 117(2):173–229, 2000.
- [3] K.D. Forbus, K. Carney, B.L. Sherin, and L.C. Ureel. Vmodel: A visual qualitative modeling environment for middle-school students. *AI Magazine*, 26(3):63–72, 2005.
- [4] K. Leelawong and G. Biswas. Designing learning by teaching agents: The betty’s brain system. *International Journal of Artificial Intelligence in Education*, 18(3):181–208, 2008.
- [5] P. Struss and C. Price. Model-based systems in the automotive industry. *AI Magazine*, 24:17–34, 2003.

# Credibility-limited Revision Operators in Propositional Logic<sup>1</sup>

Richard Booth <sup>a</sup>

Eduardo Fermé <sup>b</sup>

Sébastien Konieczny <sup>c</sup>

Ramón Pino Pérez <sup>d</sup>

<sup>a</sup> *University of Luxembourg, Luxembourg*

<sup>b</sup> *Universidade da Madeira, Funchal, Portugal*

<sup>c</sup> *CRIL - CNRS, Lens, France*

<sup>d</sup> *Universidad de Los Andes, Merida, Venezuela*

## 1 Introduction

In Belief Revision the new information is generally accepted, following the principle of primacy of update (success postulate). In some cases this behavior can be criticized and one could require that some new pieces of information can be rejected by the agent because for instance of insufficient plausibility. This has given rise to several approaches of non-prioritized Belief Revision. (For an overview see [2].) Among these approaches one can note the family of operators defined in [3], called *credibility-limited revision operators*, where a successful revision is obtained only if the new information is a formula that belongs to a set of credible formulas.

When the pieces of information of the system are encoded using propositional logic, the AGM framework can be simplified, as shown by [4]. In this particular case both the beliefs of the agent and the new evidence are represented by a propositional formula. Katsuno and Mendelzon (hereafter KM) also proposed a representation theorem in terms of plausibility pre-orders on interpretations (faithful assignment).

In this work we study credibility-limited revision operators when the pieces of information are represented in propositional logic. We propose a set of postulates and a representation theorem for credibility-limited revision operators. Then we explore how to generalize these definitions to Iterated Belief Revision operators, using epistemic states, in the Darwiche and Pearl style [1].

## 2 Credibility-limited revision in the KM framework

We first define credibility-limited revision operators via postulates.

**Definition 1** *A binary operator  $\circ$  over propositional formulas satisfying P1-P6 below will be called a CL (Credibility-Limited) revision operator:*

- |   |                               |
|---|-------------------------------|
| <b>(P1)</b> $\varphi \circ \alpha \vdash \alpha$ or $\varphi \circ \alpha \equiv \varphi$   | <b>(Relative success)</b>     |
| <b>(P2)</b> If $\varphi \wedge \alpha \not\vdash \perp$ then $\varphi \circ \alpha \equiv \varphi \wedge \alpha$  | <b>(Vacuity)</b>              |
| <b>(P3)</b> $\varphi \circ \alpha \not\vdash \perp$   | <b>(Strong coherence)</b>     |
| <b>(P4)</b> If $\varphi \equiv \psi$ and $\alpha \equiv \beta$ then $\varphi \circ \alpha \equiv \psi \circ \beta$  | <b>(Syntax independence)</b>  |
| <b>(P5)</b> If $\varphi \circ \alpha \vdash \alpha$ and $\alpha \vdash \beta$ then $\varphi \circ \beta \vdash \beta$   | <b>(Success monotonicity)</b> |
| <b>(P6)</b> $\varphi \circ (\alpha \vee \beta) \equiv \begin{cases} \varphi \circ \alpha \text{ or} \\ \varphi \circ \beta \text{ or} \\ (\varphi \circ \alpha) \vee (\varphi \circ \beta) \end{cases}$ | <b>(Trichotomy)</b>           |

<sup>1</sup>This paper was presented at the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012), June 2012.

The semantic construction of these operators is then given as follows. (Note that  $\llbracket \varphi \rrbracket$  denotes the set of models of  $\varphi$ .)

**Definition 2** A *CL faithful assignment* (CLF-assignment for short) is a function mapping each consistent formula  $\varphi$  into a pair  $(C_\varphi, \leq_\varphi)$  where  $\llbracket \varphi \rrbracket \subseteq C_\varphi \subseteq \mathcal{V}$ ,  $\leq_\varphi$  is a total pre-order on  $C_\varphi$ , and the following conditions hold for all  $\omega, \omega' \in C_\varphi$ :

1. If  $\omega \models \varphi$  and  $\omega' \models \varphi$ , then  $\omega \simeq_\varphi \omega'$
2. If  $\omega \models \varphi$  and  $\omega' \not\models \varphi$ , then  $\omega <_\varphi \omega'$
3. If  $\varphi \equiv \varphi'$ , then  $(C_\varphi, \leq_\varphi) = (C_{\varphi'}, \leq_{\varphi'})$

We obtain the following representation theorem for CL revision operators in the propositional setting.

**Theorem 1**  $\circ$  is a CL revision operator iff there exists a CLF-assignment  $\varphi \mapsto (C_\varphi, \leq_\varphi)$  such that

$$\llbracket \varphi \circ \alpha \rrbracket = \begin{cases} \min(\llbracket \alpha \rrbracket, \leq_\varphi) & \text{if } \llbracket \alpha \rrbracket \cap C_\varphi \neq \emptyset \\ \llbracket \varphi \rrbracket & \text{otherwise} \end{cases}$$

### 3 Credibility-limited iterated revision operators

When moving to the iterated case, we follow Darwiche and Pearl in assuming agents start from a more comprehensive *epistemic state* rather than a simple propositional formula, and that revision is a function (epistemic state, formula)  $\mapsto$  (epistemic state). To any epistemic state  $\Psi$  we can associate a formula  $B(\Psi)$  representing the beliefs held in  $\Psi$ . We define the class of iterated CL revision operators we are interested in via postulates inspired by those of Darwiche and Pearl.

**Definition 3** A *CLIR (Credibility-limited Iterated Revision) operator* is a revision operator  $\circ$  satisfying the following four properties (where  $C_\Psi \stackrel{\text{def}}{=} \{\beta \mid B(\Psi \circ \beta) \vdash \beta\}$ ):

- (CLDP1) If  $\alpha \vdash \mu$  and  $\alpha \in C(\Psi)$ , then  $B((\Psi \circ \mu) \circ \alpha) \equiv B(\Psi \circ \alpha)$
- (CLDP2) If  $\alpha \vdash \neg \mu$  and  $\alpha, \mu \in C(\Psi)$ , then  $B((\Psi \circ \mu) \circ \alpha) \equiv B(\Psi \circ \alpha)$
- (CLP) If  $B(\Psi \circ \alpha) \not\vdash \neg \mu$  and  $\alpha, \mu \in C(\Psi)$ , then  $B((\Psi \circ \mu) \circ \alpha) \vdash \mu$
- (CLCD) If  $\alpha \vdash \neg \mu$ ,  $\alpha \notin C(\Psi)$  and  $\mu \in C(\Psi)$ , then  $\alpha \notin C(\Psi \circ \mu)$ .

Now let us define the assignments corresponding to CLIR operators.

**Definition 4** Let  $\circ$  be a revision operator on epistemic states and let  $\Psi \mapsto (C_\Psi, \leq_\Psi)$  be a CLF-assignment as in Definition 2. This assignment will be called an *iterable CL faithful assignment* (ICLF-assignment for short) if it satisfies the following properties:

- (CR0) If  $\omega \in C_\Psi$  and  $\llbracket \alpha \rrbracket \cap C_\Psi \neq \emptyset$ , then  $\omega \in C_{\Psi \circ \alpha}$
- (CR1) If  $\omega, \omega' \in \llbracket \alpha \rrbracket \cap C_\Psi$  then  $\omega \leq_\Psi \omega'$  iff  $\omega \leq_{\Psi \circ \alpha} \omega'$
- (CR2) If  $\omega, \omega' \in \llbracket \neg \alpha \rrbracket \cap C_\Psi$  and  $\llbracket \alpha \rrbracket \cap C_\Psi \neq \emptyset$  then  $\omega \leq_\Psi \omega'$  iff  $\omega \leq_{\Psi \circ \alpha} \omega'$
- (CR3) If  $\omega \in \llbracket \alpha \rrbracket \cap C_\Psi$ ,  $\omega' \notin C_\Psi$ ,  $\omega' \in \llbracket \alpha \rrbracket$  and  $\omega, \omega' \in C_{\Psi \circ \alpha}$  then  $\omega <_{\Psi \circ \alpha} \omega'$
- (CR4) If  $\omega \in \llbracket \neg \alpha \rrbracket$ ,  $\omega \notin C_\Psi$  and  $\llbracket \alpha \rrbracket \cap C_\Psi \neq \emptyset$ , then  $\omega \notin C_{\Psi \circ \alpha}$
- (CRP) If  $\omega \in \llbracket \alpha \rrbracket \cap C_\Psi$  and  $\omega' \in \llbracket \neg \alpha \rrbracket \cap C_\Psi$  then  $\omega \leq_\Psi \omega'$  implies  $\omega <_{\Psi \circ \alpha} \omega'$

Finally we obtain the representation theorem for CLIR operators.

**Theorem 2** Let  $\circ$  be a revision operator on epistemic states and  $\Psi \mapsto (C_\Psi, \leq_\Psi)$  be a CLF-assignment as in Definition 2. Then,  $\circ$  is a CLIR operator iff  $\Psi \mapsto (C_\Psi, \leq_\Psi)$  is an ICLF-assignment.

In the full paper we also give some concrete examples of CLIR operators based on lexicographic revision and on using the Hamming distance between propositional models.

## References

- [1] A. Darwiche and J. Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89:1–29, 1997.
- [2] E. Fermé and S.O. Hansson. AGM 25 years: Twenty-five years of research in belief change. *Journal of Philosophical Logic*, 40:295–331, 2011.
- [3] S.O. Hansson, E. Fermé, J. Cantwell, and M. Falappa. Credibility-limited revision. *Journal of Symbolic Logic*, 66:1581–1596, 2001.
- [4] H. Katsuno and A. O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.

# Linkage Neighbors, Optimal Mixing and Forced Improvements in Genetic Algorithms

Peter A. N. Bosman<sup>a</sup>

D. Thierens<sup>b</sup>

<sup>a</sup> *Centrum Wiskunde & Informatica, Amsterdam, The Netherlands, Peter.Bosman@cwi.nl*

<sup>b</sup> *Utrecht University, Utrecht, The Netherlands, Dirk.Thierens@cs.uu.nl*

---

The full version of this paper has been accepted for publication at the *Genetic and Evolutionary Computation Conference (GECCO-2012)*.

## 1 Introduction

Linkage learning is a key research topic in evolutionary computation, the goal of which is to identify, during optimization, key linkage relations between variables. This means that, for the sake of efficient optimization, these variables should be considered jointly when generating new solutions (by exchanging parts between previously evaluated solutions). Good, linkage-friendly variation operators can be engineered for specific problems. The complexity of real-world problems however typically requires taking a perspective of black-box optimization (BBO), i.e. making no assumptions on the optimization problem. Because engineering good variation operators beforehand is then not possible, linkage learning becomes of great importance. A very recent EA that learns and exploits linkage is the Linkage Tree Genetic Algorithm (LTGA). LTGA exhibits excellent performance on several benchmark problems, outperforming a variety of well-known EAs. The LT linkage model has great value as it can be learned from data efficiently via hierarchical clustering and can represent variable linkages at different levels, from the smallest sets (i.e. singletons) of linked variables to the set of all variables. It however also has drawbacks. An LT is always fully built at every generation and fully traversed in the generation of every new solution. Moreover, an LT model has limited ability to represent overlapping building blocks. We therefore propose a different type of linkage model that is more flexible: the Linkage Neighbors (LN). The LN can naturally represent overlapping building blocks and can be learned from data at least as efficiently as an LT, i.e. in  $\mathcal{O}(nl^2)$  time where  $l$  is the number of problem variables and  $n$  is the population size. Furthermore, learning a LN can be parallelized in a straightforward manner. In this abstract we focus on the LN model. In the full version of this paper, in addition the use of forced improvements is described which is used in a recent efficient type of EA named Genepool Optimal Mixing Evolutionary Algorithm (GOMEA). LTGA is also of the GOMEA type.

## 2 Linkage Neighbors

The concept of the Linkage Neighbors (LN) model is to focus on each problem variable separately and determine its (nearest) neighbors in terms of linkage. Thus, for each variable  $X_i$  there is a set that contains  $X_i$  and its linkage neighbors. The potential to model overlapping building blocks with the LN is greater than with the LT. For instance, for two building blocks consisting of variable indices  $[0, 1, 2]$  and  $[1, 2, 3]$  respectively, the linkage tree can only represent this overlap by putting all variables together, i.e.  $[0, 1, 2, 3]$  and at a lower level also include either  $[0, 1, 2]$  and  $[3]$  or  $[0]$  and  $[1, 2, 3]$ . In any case, indices 1 and 2 can only appear in one of either sets. In the LN model however, overlap can be represented by  $[0, 1, 2]$ ,  $[1, 2, 0, 3]$ ,  $[2, 1, 0, 3]$  and  $[3, 2, 1]$  where in each set the second to last variable are linkage neighbors of the first variable. In addition to the increased ability to represent overlap, there is also a conceptual motivation for the LN model. In optimization, each problem variable needs to be set in a certain manner. To set this value correctly, the values for its linkage neighbors need to be taken into account. Therefore, for each variable we need to know its linkage neighbors. To learn an instance of the Linkage Neighbors model, statistical hypothesis testing can be used, which, in case of the well-known likelihood-ratio test amounts to using a threshold on the mutual information between pairs of variables. The use of such an independence threshold  $\theta^{MI-indep}$  provides a first, base  $\mathcal{O}(nl^2)$  learning algorithm for the LN model: for each variable  $X_i$ , determine its linkage neighbors as all  $X_j$ ,  $i \neq j$  for which  $MI(X_i, X_j) \geq \theta^{MI-indep}$ . In the full version of this paper, three other learning algorithms with the same runtime complexity are proposed.

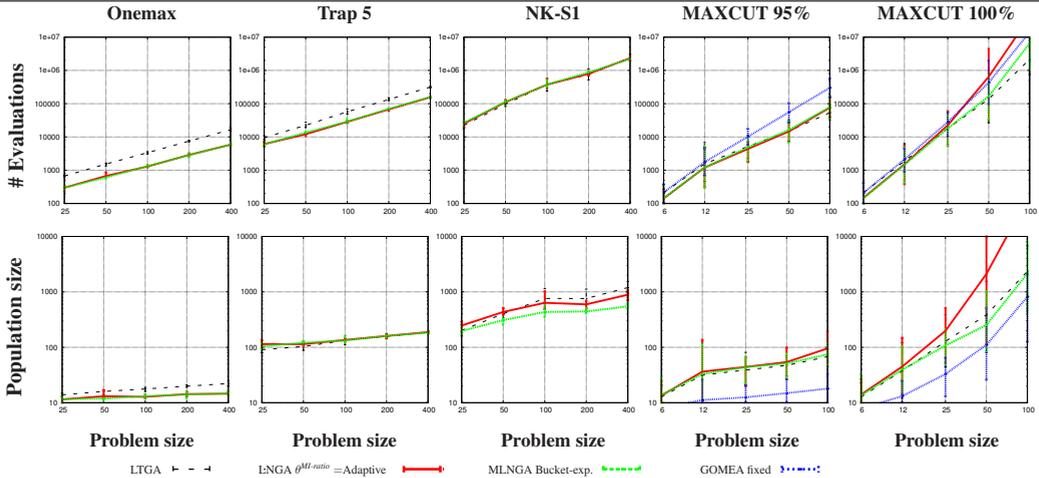


Figure 1: Scalability of LTGA, LNGA, MLNGA and GOMEA-fixed variants on all problems.

### 3 Multiscale Linkage Neighbors

Earlier experimental results indicate that modeling linkages at multiple scales is important. First results obtained using the LN model are consistent with these results. Although the hierarchical structure as used in the LTGA is a natural manner of doing so, it is still limited in its ability to represent overlapping dependencies. We therefore consider a multiscale variant of the LN model. To do so, we consider a decomposition of the LN model. In other words, as a basis we consider the same approach as in the LNGA using  $\theta^{MI-indep}$ . We then progressively subdivide the accepted linkage neighbors. We consider 4 alternative approaches that all have the same computational complexity, all of which are described in detail in the full version of this paper. The most successful approach, called bucket-exponential decomposition, decomposes the linkage neighborhood of each variable  $X_i$  by first bucket-sorting the neighborhood variables on the basis of their mutual information values with  $X_i$ . To create the final decomposition, the buckets are regarded in order of decreasing mutual information values, but a new subset is only introduced if the number of variables in all prior buckets is at least twice the number of variables in the previously added subset in the decomposition.

### 4 Summarized Results and Conclusions

The performance of the LN-GOMEA with forced improvements, which, for short, we will refer to as LNGA (Linkage Neighbors Genetic Algorithm), as well as MLN-GOMEA with forced improvements, which, for short, we will refer to as MLNGA (Multiscale Linkage Neighbors Genetic Algorithm), on a set of well-known linkage benchmark problems is shown in Figure 1. In addition, a well-known combinatorial optimization problem called weighted MAXCUT is considered. The goal in weighted MAXCUT is to split the set of vertices of a given weighted graph into two sets such that the combined weight of all edges that are thereby cut, i.e. running between vertices in different sets, is maximized. Only the results for the best variants are shown. For weighted MAXCUT, results obtained with an a priori-fixed neighborhood, namely all pairs of variables (because the graphs in the considered instances are all fully connected), are also shown.

The finally selected variants of LNGA and MLNGA exhibit excellent performance on the linkage benchmark problems, performing similar or better than LTGA in terms of population size and number of evaluations. On weighted MAXCUT however, when targeting the true optimum, the LN model is clearly less efficient. Moreover, on this problem LTGA outperforms MLNGA as the problem size increases. The scale-up of MLNGA is quite similar in shape to that of the GOMEA with the fixed linkage structure containing only all variable pairs. The worse behavior, compared to LTGA, of MLNGA may therefore well be a result of the large number of small linkage sets in the MLN model.

Although more flexible than the LT model, the LN and MLN models still admit learning algorithms with efficient  $\mathcal{O}(nl^2)$  runtime complexity. Moreover, the proposed algorithms perform binning of mutual information values independently for each variable, making parallelization straightforward. Although this makes the LN and MLN models interesting to study further, results on weighted MAXCUT indicate that the LT model still offers a superior way of configuring a multi-scale linkage model. Therefore, it might alternatively be interesting to study the use of concepts introduced in this paper to adjust the LT model.

# Modelling Collective Decision Making in Groups and Crowds (extended abstract)

Tibor Bosse, Mark Hoogendoorn, Michel C.A. Klein,  
Jan Treur, C. Natalie van der Wal, Arlette van Wissen

*VU University Amsterdam, Agent Systems Research Group,  
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands*

## 1 Introduction

Behavioural patterns emerging in large crowds are often difficult to regulate. Various examples have shown how things can easily get out of control when many people come together during big events. Especially within crowds, the consequences can be devastating when emotion spirals (e.g., for aggression or fear) develop to high levels. An example of such an emergent pattern was seen on Dam square in Amsterdam on the 4th of May in 2010, when large numbers gathered for the national remembrance of the dead ('dodenherdenking') [8]. In the middle of a two-minute period of silence, one person started shouting, causing panic to occur among the people present. As a result of a panic spiral, people fled in different directions. Eventually, the seriousness of the event turned out to be relatively mild, since 'only' a number of persons ended up in hospitals with fractures.

## 2 The ASCRIBE Model

In order to understand and describe the patterns mentioned in the introduction, agent-based modelling techniques can be used. Once an accurate computational model of such phenomena has been designed, it can for instance be used to perform simulations of various scenarios, but also to test certain measures to avoid unwanted emergent patterns (e.g. manipulations with ambient devices). As a first step in this process, a biologically plausible agent-based model has been developed called ASCRIBE (for Agent-based Social Contagion Regarding Intentions Beliefs and Emotions) [5]. Within this model (inspired by theories from Social Neuroscience, e.g. [1,6,7]), behaviour of agents is determined by cognitive as well as affective states. Cognitive states include beliefs and intentions of agents, whereas affective states include emotions such as fear, but also positive emotions related to actions: for example, going to a place believed to be safe. Also the interplay between these different cognitive and affective states is captured in the model. On the one hand these internally interacting states are individual, private states, but on the other hand they are easily affected by similar states of other persons via verbal and/or nonverbal inter-person interaction (sometimes called *contagion* of mental states). In addition, the ASCRIBE model takes into account a number of parameters representing personal characteristics for each agent (e.g., expressivity of emotions). The ASCRIBE model is formalised in a numerical manner: all cognitive and affective states are represented in terms of real numbers between 0 and 1, and all dependencies between states in terms of dynamical mathematical equations.

### 3 Case Study: the May 4 Incident

In the work described in the full version of this paper [2], a next step is taken, namely to use the ASCRIBE model to describe an actual event. For this purpose, the incident on Dam square on May 4th is chosen. As a first step in this process, useful empirical data has been extracted from available video material and witness reports. In order to specialise the existing agent-based model to this case, values for most of the parameters of the model were set by hand at default values, whereas values of other parameters were automatically tuned by use of a parameter tuning method developed earlier [3]. To make this parameter tuning possible, the predictions of the ASCRIBE model were compared with the actual empirical data, and based upon the sensitivity of the overall accuracy upon changes in parameter values the values were updated. By comparing different default settings for the hand-set parameters relating to contagion of emotions, beliefs and intentions, it was possible to analyse the contribution of contagion in the model: parameter settings indicating low or no contagion show higher deviations from the empirical data. By using a formal error measure, the difference in performance between the tuned model with and without contagion was found to be 18%. This illustrates that emotion contagion is an important component in crowd behaviour. In addition, for the case study, the model with contagion was shown to perform significantly better than the well-known Helbing model [4], one of the most influential models in the area of crowd simulation.

### References

- [1] Bechara, A. and Damasio, A. The Somatic Marker Hypothesis: a neural theory of economic decision. *Games and Economic Behavior* 52, pp. 336-372, 2004.
- [2] Bosse, T., Hoogendoorn, M., Klein, M.C.A., Treur, J., Wal, C.N. van der, and Wissen, A. van. Modelling Collective Decision Making in Groups and Crowds: Integrating Social Contagion and Interacting Emotions, Beliefs and Intentions. *Autonomous Agents and Multi-Agent Systems Journal*, 2012, in press.
- [3] Bosse, T., Memon, Z.A., Treur, J., and Umair, M. An Adaptive Human-Aware Software Agent Supporting Attention-Demanding Tasks. In: Yang, J.-J., Yokoo, M., Ito, T., Jin, Z., Scerri, P. (eds.), *Proceedings of the 12th International Conference on Principles of Practice in Multi-Agent Systems, PRIMA'09*, Lecture Notes in AI, vol. 5925, pp. 292--307. Springer Verlag, Heidelberg, 2009.
- [4] Helbing, D., Farkas, I., Vicsek, T. Simulating Dynamical Features of Escape Panic. *Nature*, vol. 407, issue 6803, 2000, pp. 487-490.
- [5] Hoogendoorn, M., Treur, J., Wal, C.N. van der, and Wissen, A. van. Modelling the Interplay of Emotions, Beliefs and Intentions within Collective Decision Making Based on Insights from Social Neuroscience. In: Wong, K.K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.), *Proceedings of the 17th International Conference on Neural Information Processing, ICONIP'10*. Lecture Notes in Artificial Intelligence, vol. 6443. Springer Verlag, 2010, pp. 196-206.
- [6] Iacoboni, M. *Mirroring People*. Farrar, Straus & Giroux, New York, 2008.
- [7] Pineda, J.A. (ed.). *Mirror Neuron Systems: the Role of Mirroring Processes in Social Cognition*. Humana Press Inc, New Jersey, 2009.
- [8] [http://en.wikipedia.org/wiki/Remembrance\\_of\\_the\\_Dead](http://en.wikipedia.org/wiki/Remembrance_of_the_Dead)

# Reinforcement Learning Transfer via Sparse Coding<sup>1</sup>

Haitham Bou Ammar<sup>a</sup>   Karl Tuyls<sup>a</sup>   Matthew E. Taylor<sup>b</sup>  
Kurt Driessens<sup>a</sup>   Gerhard Weiss<sup>a</sup>

<sup>a</sup> *Department of Knowledge Engineering, Maastricht University, The Netherlands*

<sup>b</sup> *Departement of Computer Science, Lafayette College, USA*

## 1 Introduction

Although reinforcement learning (RL) has been successfully deployed in a variety of tasks, learning speed remains a fundamental problem when applying RL in complex environments. Transfer learning aims to ameliorate this shortcoming by speeding up learning through the adaptation of previously learned behaviors in similar tasks. Transfer techniques often use an inter-task mapping. Instead of relying on a hand-coded inter-task mapping, this paper proposes a novel transfer learning method capable of autonomously creating an inter-task mapping by using a novel combination of sparse coding, sparse projection learning and sparse Gaussian processes. Experiments not only show successful transfer of information between similar tasks, inverted pendulum to cart pole, but also between two very different domains: mountain car to cart pole. This paper shows that the learned inter-task mapping can (1) improve the performance of the policy learned for a fixed sample size, (2) reduce the learning times needed by the algorithms to converge to a policy for a fixed sample size (not shown in this abstract), and (3) converge faster to a near-optimal policy given a large sample size.

## 2 The approach

To establish transfer without a given inter-task mapping, the technique uses a multi-step process: first, an inter-task mapping is induced automatically, then, samples obtained a good source policy are mapped to the target task improving the quality of samples made available to an off-line RL algorithm. Learning the inter-task mapping is cast as a supervised learning problem. We define the inter-task mapping to be,  $\chi : S_s \times A_s \times S_s \rightarrow S_t \times A_t \times S_t$ , where  $S$  and  $A$  represent the state and action spaces for each of the source  $s$  and target  $t$  task respectively.

To build a data-set of matched  $(s, a, s)$ -triplets usable by the supervised learning algorithm, we use sparse coding to map the triplets of both tasks to an information rich high dimensional feature space using Sparse Coding [3] (SC). Given a set of  $m$   $k$ -dimensional vectors,  $\zeta$ , SC aims to find a set of  $n$  basis vectors,  $\mathbf{b}$ , and activations,  $a$ , with  $n > k$ . SC solves the following optimization problem,

$$\min_{\{\mathbf{b}_j\}, \{a_j^{(i)}\}} \sum_{i=1}^m \frac{1}{2\sigma^2} \|\zeta^{(i)} - \sum_{j=1}^n \mathbf{b}_j a_j^{(i)}\|_2^2 + \beta \sum_{i=1}^m \sum_{j=1}^n \|a_j^{(i)}\|_1$$

*s.t.*  $\|\mathbf{b}_j\|_2^2 \leq c, \forall j = \{1, 2, \dots, n\}$

Given random source and target task samples we proceed by: (1) SC the source samples to the target dimensions, (2) SC code the attained bases and activations to produce a rich feature space, (3) projecting the target task samples into the sparse coded space, (4) collecting samples using a similarity measure, and (5) use regression to approximate  $\chi$ .

<sup>1</sup>Accepted at the eleventh international conference on autonomous agents and multiagent systems (AAMAS), Valencia, 2012.

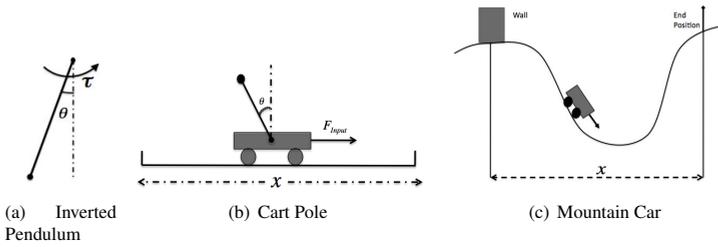


Figure 1: Experimental domains

### 3 Experiments and Results

We conducted experiments in two transfer settings to evaluate the framework. The first was the transfer from the Inverted Pendulum (IP), Figure 1(a), to the Cart Pole (CP), Figure 1(b). The second experiment transfers between Mountain Car (MC), Figure 1(c) and CP. We performed experiments using two different sample based reinforcement learning method in the target, Least Square Policy Iteration [2] and Fitted-Q-Iteration [1]. As demonstrated, the approach is capable of automatically learning an inter-task mapping and the transfer is capable of improving: (1) Jump-start, (2) learning speed, and (3) convergence times.

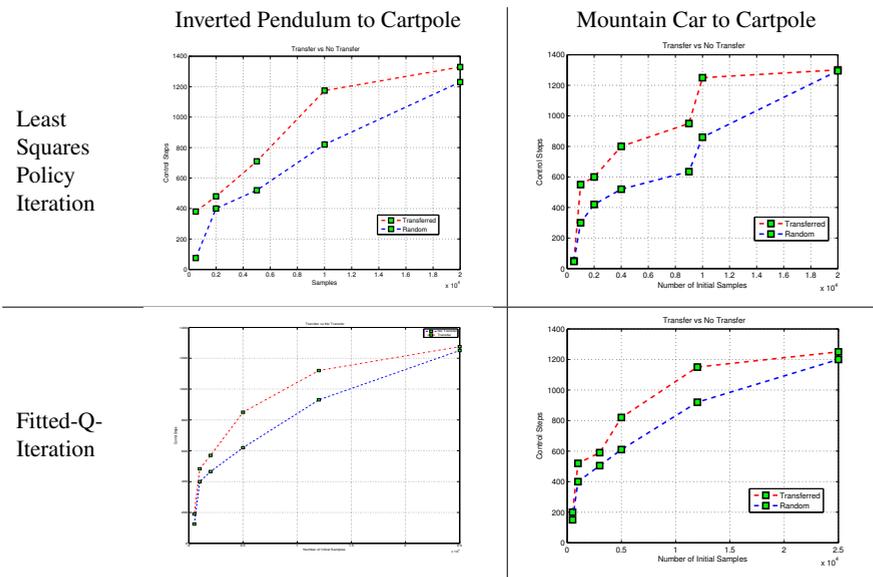


Figure 2: Transfer Learning results using two off-line RL algorithms in two different transfer scenario's. Performance is measured by the number of control steps the pole is in an upright position.

### References

- [1] L. Buşoniu, R. Babuška, B. De Schutter, and D. Ernst. *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC Press, Boca Raton, Florida, 2010.
- [2] Michail G. Lagoudakis and Ronald Parr. Least-squares policy iteration. *J. Mach. Learn. Res.*, 4:1107–1149, December 2003.
- [3] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In *In NIPS*, pages 801–808. NIPS, 2007.

# Recognizing Demand Patterns from Smart Card Data for Agent-Based Micro-simulation of Public Transport <sup>1</sup>

Paul Bouman

Milan Lovric

Ting Li

Evelien van der Hurk

Leo Kroon

Peter Vervest

{PBouman,MLovric,TLi,EHurk,LKroon,PVervest}@rsm.nl  
Rotterdam School of Management, Erasmus University  
P.O. Box 1738, 3000 DR, Rotterdam, The Netherlands

**Introduction** In public transport systems without seat reservations, the question of how fluctuating demand can be serviced in a cost-efficient way poses a major challenge. Peaks in demand have a high toll on the costs, since they dictate the required amount of staff and the number of vehicles, while vehicles that are almost empty generate a net loss for the operator. Tools that allow the public transport operator to evaluate the effects of operational and strategic decisions on costs and demand are therefore vital to achieve the goal of improving the service quality and financial performance. However, most of the tools used in practice aggregate the passengers to homogeneous flows, either because detailed data is not available, or to reduce the complexity the decision maker has to face.

During recent years, smart card systems have been introduced that log all movements of individual passengers through the systems. This gives a lot of detailed data that was previously unavailable. However, given the body of research related to smart card data, we can see that incorporating such data into the tools used for decision making is a non-trivial task [3].

In agent-based micro simulation, individual passengers and vehicles are modeled through agents that interact with the public transport system according to their individual goals. This allows us to model demand on an individual level, which gives a lot of flexibility. In this paper, we will use the MATSim simulation package [1] to simulate an urban scenario, based on real smart card data. We present an algorithm that allows us to detect travel patterns in raw smart card data and turn these patterns into activity based demand suitable for MATSim. Within MATSim, all agents have to perform activities and after each simulation they try to adapt their travel plans in such a way that their utility is improved.

We limit our field of application to the study of *revenue management* [4]. Three different pricing structures are evaluated in our model. We present the results of our simulation and discuss how these results can be improved in the future. Additionally, we discuss a number of research directions that are necessary in order to create useful and reliable decision support tools based on the combination of smart card data and agent-based simulation.

**Demand** When we analyze the smart card data, the number of observed journeys differs a lot between individual passengers. In order to deal with this, we introduce three types of demand: *trip-based*, *tour-based* and *pattern-based* demand.

To derive these three types of demand from our dataset of 23 million journeys, we bundle the journeys according to their smart card id and sorted the journeys according to the time they were performed. We then pick a day for which we want to extract demand. We generate demand for each traveller during this chosen day. First, we check whether the passenger is a commuter. If he or she has a sufficient number of journeys  $\theta$  in the complete dataset, we take the two most frequently visited stations. If these two frequent stations occur

---

<sup>1</sup>This paper was published in *Proceedings of the Seventh International Workshop on Agents in Traffic and Transportation*, 2012. Editors: Matteo Vasilirani, Fraziska Klügl, Eduardo Camponogara, and Hiromitsu Hattori.

in the majority of the journeys, we calculate the time the individual spends at each of these stations. We then denote the stations where most time is spent as “home” and the other as “work”. We can then calculate the mean and standard deviation of arrival, departure and duration observed at the work station and generate *pattern-based* demand. If we can’t find such patterns, we check whether somebody travelled a tour during this day. If this is the case, we can generate *tour-based* demand. Otherwise, we generate *trip-based* demand.

We applied this demand generation procedure for different values of the minimum number of journeys  $\theta$ . Since the number of individuals in the *pattern-based* category dropped quickly, we ran the simulation for three populations of  $\theta = 80$ ,  $\theta = 120$  and  $\theta = \infty$ . For our pricing strategy, we took a price inspired by the real world pricing policies. In our experiments, we simulated the system using the default settings of the 0.3.0 MATSim package (which contains [2] as a default utility function), on each of our populations through the network two times: once with a single tariff over the full day and once with a discount outside the peak hours (the peak hours are between 7:00–9:00 and 16:00–19:00).

**Results** The results are presented in Figure 2 of the paper. In the case where we have a single tariff, a move from the  $\theta = 80$  case to the  $\theta = 120$  case results in the morning peak becoming a bit smaller and the evening peak becoming a plateau that is a bit wider. This implies that some passengers who were pattern based in the  $\theta = 80$  case that turned to tour or trip-based in the  $\theta = 120$  case, tend to move away from the morning peak towards the evening peak. When we increase  $\theta$  to  $\infty$ , we see that the morning peak increases a bit and the evening peak increases a lot. This suggests that some of the pattern-based agents in the  $\theta = 120$  case actually traveled during the morning peak in the  $\theta = \infty$  case, where they were less flexible. One drawback is that the simulation generates a sharp peak at 5:00 (when the first vehicles start to drive) and 24:00 (when the simulation ends), which seems to be an artifact of the current model.

When discounts are added, new peaks emerge just outside undiscounted periods. Even with a very small discount, most of the agents have an incentive to divert from their initial plans. There can be two reasons for this behavior: either the agent is flexible enough to divert without losing utility, or the disutility of being early or late is smaller than the utility gained from the discount. If we compare the  $\theta = 80$  with the  $\theta = 120$  case, a difference can be observed in the patterns that emerge within the peak-hour time windows. The evening peak in the  $\theta = 120$  case has a triangular structure, when it compared to the  $\theta = 80$  case. When we increase  $\theta$  to  $\infty$ , we get this triangular pattern in the morning peak as well and the effect in the evening peak is amplified.

**Discussion and Future Research** These results suggest multiple directions for future research. One thing that can be improved, is *demand generation*. Our pattern recognition approach is very basic and could be improved in order to detect more intricate patterns. Also, it could make sense to exploit the correlations of the time of arrival, departure and the duration of being at the home/work stations.

Our model also needs some form of *calibration*. In the current model, too many agents divert from their observed travel patterns. To achieve this, we may need new utility models, that include a more heterogeneous sensitivity to the price of a journey. Price elasticities of individual agents could be calibrated using survey data and discrete choice methods.

When the model is calibrated, the next step is to *validate* the model. For this, we could split up the dataset into two parts, where the first part is used to generate demand and the second part is used for validation. Another approach would be to collect smart card data before and after an actual policy change and see how well the model predicts the outcome.

## References

- [1] Multi-agent transport simulation toolkit, 2012. <http://www.matsim.org>.
- [2] David Charypar and Kai Nagel. Generating complete all-day activity plans with genetic algorithms. *Transportation*, 32:369–397, 2005. 10.1007/s11116-004-8287-y.
- [3] Marie Pier Pelletier, Martin Trépanier, and Catherine Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557 – 568, 2011.
- [4] K.T. Talluri and G. Van Ryzin. *The theory and practice of revenue management*, volume 68. Springer Verlag, 2005.

# Solving Satisfiability in Fuzzy Logics with Evolution Strategies

Tim Brys <sup>a</sup>   Yann-Michaël De Hauwere <sup>a</sup>   Martine De Cock <sup>b</sup>   Ann Nowé <sup>a</sup>

<sup>a</sup> *Computational Modeling Lab, VUB, Pleinlaan 2, B-1050 Brussels*

<sup>b</sup> *Dept. of Applied Math. and Comp. Sc., UGent, Krijgslaan 281 (S9), B-9000 Gent*

## Abstract

Satisfiability in propositional logic is well researched and many approaches to checking and solving exist. In infinite-valued or fuzzy logics, however, there have only recently been attempts at developing methods for solving satisfiability. In this paper, we propose a new incomplete solver, based on a class of continuous optimization algorithms called evolution strategies. We show experimentally that our method is an important contribution to the state of the art in incomplete fuzzy-SAT solvers.

## 1 Introduction

A logical formula, or a set of formulas, is said to be satisfiable if there exists a truth assignment to its variables that makes every formula true. Satisfiability checking is verifying whether such an assignment exists, and satisfiability solving means finding such an assignment. This problem is known as SAT in propositional logic and is of interest to researchers from various domains, as many problems can be reformulated as a SAT problem and subsequently solved by a state-of-the-art SAT solver.

In fuzzy logics, the same principle of satisfiability exists,  $SAT_\infty$ , and, like its classical counterpart, is useful for solving a variety of problems. Indeed, many fuzzy reasoning tasks can be reduced to  $SAT_\infty$ , including reasoning about vague concepts in the context of the semantic web, fuzzy spatial reasoning and fuzzy answer set programming.

Solving satisfiability in fuzzy logics has however received much less attention than its counterpart in classical logics. Schockaert et al. propose a solver in [3] which reduces the infinite-valued logic to a finite-valued one and then applies a constraint satisfaction solver to check satisfiability, iteratively refining the discretization until a solution is found. This discretization makes the approach ineffective on certain classes of problems, where the satisfaction bounds for formulas are very fine-grained.

In this paper [1], we consider satisfiability checking and solving as an optimization problem in a continuous domain. We propose an incomplete solver capable of deciding  $SAT_\infty$  but not  $UNSAT_\infty$ , based on the state-of-the-art algorithm in evolution strategies CMA-ES[2].

## 2 Optimizing $SAT_\infty$

As we will investigate an optimization approach to solving satisfiability in fuzzy logics, we need to reformulate  $SAT_\infty$  instances as optimization problems, i.e. defining a function over the solution space such that optimizing this function corresponds to solving the  $SAT_\infty$  instance. A  $SAT_\infty$  problem consists of a set  $\Theta$  of fuzzy formulas  $\alpha_i$ , each of which must be satisfied to a certain degree, as defined by an upper and lower bound per formula (the upper bound is usually set to 1). Given these  $n$  formulas  $\alpha_i$ , bounds  $(u_i, l_i)$ , and an interpretation  $\mathcal{I}$  – a value assignment to the variables –, we define the objective function  $f$  as follows:

$$f(\mathcal{I}) = \frac{\sum_i^n f_{\mathcal{I}}(\alpha_i)}{n} \quad (1)$$

and, for  $1 \leq i \leq n$ ,

$$f_{\mathcal{I}}(\alpha_i) = \begin{cases} 1 & \text{if } l_i \leq [\alpha_i]_{\mathcal{I}} \leq u_i \\ \frac{[\alpha_i]_{\mathcal{I}}}{l_i} & \text{if } [\alpha_i]_{\mathcal{I}} < l_i \\ \frac{1-[\alpha_i]_{\mathcal{I}}}{1-u_i} & \text{if } [\alpha_i]_{\mathcal{I}} > u_i \end{cases} \quad (2)$$

with  $[\alpha_i]_{\mathcal{I}}$  representing the degree of satisfaction of formula  $\alpha_i$  under interpretation  $\mathcal{I}$ . Each  $f_i$  is a trapezoid function, with a plateau of value 1 when formula  $\alpha_i$ 's degree of satisfaction lies between the given bounds, and a slope leading to the plateau when the satisfaction lies outside these bounds.

This function is formulated such that the global maximum will always have a function value 1 if the  $SAT_{\infty}$  instance is satisfiable. In that case, every global maximum corresponds to a model of the problem. As CMA-ES is not guaranteed to converge to the global optimum, we have an incomplete solver, being able to decide  $SAT_{\infty}$  sometimes, but never  $UNSAT_{\infty}$ .

### 3 Experiments

We compare our solver with the state-of-the-art solver proposed by Schockaert [3], which discretizes the continuous SAT problem and solves it using a constraint satisfaction problem (CSP) solver. The results, as shown in Figures 1 and 2, indicate that on problems with a coarse granularity, i.e. bounds for formulas come from a small set (e.g.  $\mathbb{T}_4 = \{0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1\}$ ), both methods solve approximately as many instances, although the discretization approach is generally faster. On the other hand, on problems with finer, more realistic bounds ( $\mathbb{T}_{100}$ ), the performance of the discretization approach degrades much, while our solver, which handles the problem as a truly continuous problem, performs as well as on the 'easier' problems.

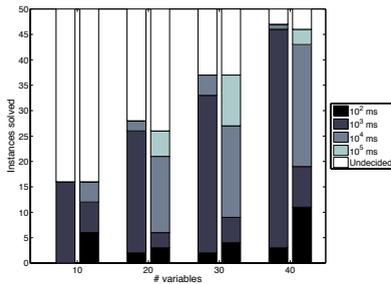


Figure 1: Results on benchmark problems from [3]. The left bars represent results with a CSP solver, right bars are CMA-ES results. Both methods perform on par in terms of the number of instances they can solve, but the CSP solver does this generally faster than our method.

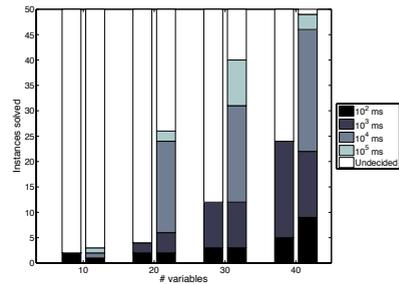


Figure 2: Results on harder problems with a higher granularity in the bounds for formulas. The left bars represent results with a CSP solver, right bars are CMA-ES results. Our approach is able to solve a significantly higher amount of problems than the CSP solver before timeout.

### References

[1] Tim Brys, Yann-Michal De Hauwere, Martine De Cock, and Ann Nowé. Solving satisfiability in fuzzy logics with evolution strategies. In *Proceedings of the 31st North American Fuzzy Information Processing Society Meeting*, 2012.

[2] N. Hansen. The CMA evolution strategy: a comparing review. In J.A. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[3] Steven Schockaert, Jeroen Janssen, and Dirk Vermeir. Satisfiability checking in Łukasiewicz logic as finite constraint satisfaction. *Journal of Automated Reasoning*, 2012.

# BNAIC

## The log-Gabor method: speech classification using spectrogram image analysis

Harm Buisman <sup>a</sup>

Eric Postma <sup>a</sup>

<sup>a</sup> *Tilburg University, Tilburg, The Netherlands*

### Abstract

In [1] we explored the suitability of the log-Gabor method, a speech analysis method inspired by [2], for automatic classification of personality and likability traits in speech. The core idea underlying the log-Gabor method is to treat the spectrogram as an image of spectro-temporal information. The image is transformed into Gabor energy values using the two-dimensional logarithmic Gabor transform, which is a standard feature extraction method in visual texture analysis. The aggregated energy values are mapped onto classes by means of a support vector machine (SVM). The log-Gabor method was tested against the Interspeech 2012 Likability and Personality Sub-Challenges [4]. The performances of the log-Gabor method are comparable to the challenge baseline. The results encourage further investigation.

## 1 Introduction

In recent years, several studies showed the use of the spectrogram in extracting perceptual cues from speech (e.g. [2, 5, 3]). As the spectrogram obtained with the short-time Fourier transform (STFT) is two-dimensional, it can be visualized and analyzed as an image, even though it is not an image in the strict sense [2]. We investigate a method that uses the two-dimensional Gabor transform. This transform analyses visual contours and features by means of spatial frequency filters that are both localized and oriented. Since the spectrogram is a time-frequency representation of speech, these filters can be interpreted as spectro-temporal filters. Our aim in this work is to determine whether the Gabor transform can be successfully applied to extract perceptual cues to personality and likability. We do this by evaluation of the method on two datasets provided by the Interspeech 2012 Likability and Personality Sub-Challenges [4]. These datasets consist of binary classes: Likable (L), Not Likable (NL) and correspondingly for personality: Openness (O/NO), Conscientiousness (C/NC), Extraversion (E/NE), Agreeableness (A/NA) and Neuroticism (N/NN). The labels were acquired by having humans rate the speech samples for their likability and (perceived) personality.

## 2 The log-Gabor method

The log-Gabor method takes a speech signal as input and in four steps generates a class estimate as output. The first step is the generation of a spectrogram using the short-time Fourier transform with variable window length. The second step is the application of the logarithmic Gabor transform, which entails filtering the spectrograms by a filterbank of logarithmic 2D-Gabor filters, which have a Gaussian shaped frequency response when plotted on a logarithmic frequency scale. The filterbank is created with filters that vary in scale and orientation in a way that minimizes filter overlap. The third step is the acquisition of raw features from the energy images (created with the logarithmic Gabor transform) by averaging over the frequency bands in the spectrogram. The raw features are normalized to the unit interval, mapped onto principal components and normalized again. The fourth and final step is the classification of the reduced and normalized features using a support vector machine with a radial basis function as kernel.

Table 1: Performance on the Interspeech 2012 Likability and Personality Sub-Challenges

Task	Baseline (%)		Performance (%)		
	Dev	Test	Dev	CV	Test
Openness	60	59	73	70	54
Conscientiousness	74	80	80	79	76
Extraversion	83	76	89	87	73
Agreeableness	68	64	73	76	62
Neuroticism	69	66	75	78	68
Personality mean	70	69	78	78	67
Likability	59	59	74	68	62

### 3 Results

As fixed parameters for the log-Gabor method we chose a filterbank of 15 scales and 8 orientations and summated the energy values over 5 frequency bands, leading to a total of 600 raw features. The variable parameters were selected in a grid search over a set of temporal resolutions, number of principal components retained and SVM parameters. In Table 1 the baselines and the log-Gabor method’s performances are presented. Dev, CV and Test represent performance on the development set, performance of 10-fold cross-validation on both development and trainingsets, and performance on the test set respectively. The baselines are the maximum performances reported in [4] acquired using random forests on 6125 features. This baseline featureset consists of 64 low-level descriptors (such as RMS energy, MFCC’s and F0) and their functionals (such as mean, median, maximum and variance). The parameters for time resolution and PCA that we used for classification of the test set were those of the best model in the cross-validation experiment.

### 4 Discussion and Conclusion

The results show that the log-Gabor method performs comparable to the baseline on the test set. What the results also show is that there is a considerable drop in performance on the test set compared to the development set. We think that this can be ascribed to overfitting on the time resolution and PCA parameters and can be reduced by using nested cross-validation for selection of these two parameters. An important question for future research is which log-Gabor features are associated with likability and personality. This could reveal how humans perceive likability and personality in speech. Another direction of research is to investigate how the log-Gabor features relate to other acoustic features, such as those in the 6125 feature baseline set. We conclude that our findings support further investigation into the log-Gabor method as a method to extract perceptual cues from speech.

### References

- [1] Harm Buisman and Eric Postma. The log-Gabor method: speech classification using spectrogram image analysis. In *Proc. Interspeech 2012*, September 2012.
- [2] Tony Ezzat, Jake Bouvrie, and Tomaso Poggio. Spectro-temporal analysis of speech using 2-D Gabor filters. In *Proc. Interspeech 2007*, volume 96, pages 1–4. Citeseer, 2007.
- [3] Bernd T. Meyer and Birger Kollmeier. Robustness of spectro-temporal features against intrinsic and extrinsic variations in automatic speech recognition. *Speech Communication*, 53(5):753–767, May 2011.
- [4] B. Schuller, S. Steidl, A. Batliner, E. Nth, A. Vinciarelli, F. Burkhardt, R. van Son, F. Wenginger, F. Eyben, T. Bocklet, G. Mohammadi, and B. Weiss. The Interspeech 2012 Speaker Trait Challenge. In *Proc. Interspeech 2012*, 2012.
- [5] Siqing Wu, Tiago H. Falk, and Wai-Yip Chan. Automatic speech emotion recognition using modulation spectral features. *Speech Communication*, 53(5):768–785, May 2011.

# Evolutionary and Swarm Computing for scaling up the Semantic Web

Christophe Guéret    Stefan Schlobach    Kathrin Dentler    Martijn Schut

*VU University Amsterdam  
De Boelelaan 1105, 1081HV Amsterdam*

## Abstract

The success of the Semantic Web, with the ever increasing publication of machine readable semantically rich data on the Web, has started to create serious problems as the scale and complexity of information outgrows the current methods in use, which are mostly based on database technology, expressive knowledge representation formalism and high-performance computing.

We argue that methods from computational intelligence (CI) can play an important role in solving these problems. In this paper we introduce and systemically discuss the typical application problems on the Semantic Web and discuss CI alternative to address the limitations of their underlying reasoning tasks consistently with respect to the increasing size, dynamicity and complexity of the data. Finally, we discuss two case studies in which we successfully applied soft computing methods to two of the main reasoning tasks; an evolutionary approach to querying, and a swarm algorithm for entailment.

*This short paper is a summary of Guéret, C.; Schlobach, S.; Dentler, K.; Schut, M.; Eiben, G. "Evolutionary and Swarm Computing for the Semantic Web". IEEE Computational Intelligence Magazine, Special Issue on Semantic Web Meets Computational Intelligence, Vol. 7, No. 2 (May 2012)*

## 1 Introduction

The World Wide Web is a decentralized system enabling the publication of documents and links between these documents on the Internet. A document is a piece of text, usually written in HTML and made available at a particular address (the URI). The links between documents are based on anchors put in these texts (hypertext links) and express a relation whose meaning depends on the interpretation made of the anchoring text. The Semantic Web uses the Web as a platform to publish and interlink data, rather than documents. This platform can then be used to build applications.

## 2 From logic proofs to optimisation problems

In their systematic analysis [3] Harmelen *et. al* argued that typical Semantic Web applications require a rather restricted set of basic reasoning tasks ("Entailment", "Consistency", "Mapping"). However, the Semantic Web combines data with Semantics, *i.e.* provide extra meaning to data, and thus yields problems related to data management as well. We therefore extend this analysis with a set of basic data manipulation tasks ("Querying", "Storage"). Table 1 contains a formal description of the different tasks Semantic Web applications have to perform, along with a short explanation of the traditional solving techniques.

Every algorithm currently being put to use on Semantic Web data has been designed as a logic based method operating over a finite set of curated triples  $T$  (a "knowledge base"). However, a typical triple set  $T$  on the Semantic Web is not static, nor curated, and all the guarantees of logical reasoning (completeness, soundness, determinism, ...) are lost. Instead, one can only *aim* at them and thus *optimize* towards these ideals. The consensual approach is to fit the Semantic Web data into a knowledge base by downloading, aggregating and curating subsets of its content. The problem is therefore adapted to fit the currently available solving methods, rather than being addressed with novel techniques. In order to find solving methods capable

Task	Formal definition	Traditional approach	Alternate formulation
Querying	Given $T$ and a query $Q$ , return the set of triples $\{t \in T\}$ such that $T \vdash t < Q$	Lookup and join	Constrained optimisation
Storage	Given $T$ and a triple $t$ return $T \cup t$	Centralized indices, Distributed Hash-tables	Clustering
Entailment	Given $T$ . Derive $t \notin T$ with $T \vdash t$	Centralized and parallelized deduction (rules)	Multi-objective optimisation
Consistency	Given $T$ . Check whether $T \vdash \perp$ (false)	Logical reasoning	Constrained optimisation
Mapping	Given $T$ and a mapping condition $c$ . Return $s, o \in T \times T$ such that $c(s, o)$ likely holds with respect to $T$	Similarities search between resources and classes. Inductive reasoning	Classification

Table 1: Tasks and traditional solving methods to make use of a set of triples  $T$ . In the table,  $\vdash$  stands for logical entailment and  $t < Q$  implies that  $t$  is an instance of  $Q$ .

of dealing with the complex character of the Semantic Web we propose to rephrase the logical formulations of the tasks as optimization problems (*c.f.* 4th column of Table 1).

Evolutionary and Swarm algorithms are known to perform well on optimization problems with large, and eventually dynamic, search spaces. We developed two use-cases leveraging these two family of algorithms for two of the tasks previously introduced: eRDF and Swarms.

**eRDF** [2] is an evolutionary computing based system solving “Querying” tasks. The queries are turned into a constraint satisfaction problem, which is in turn relaxed into a constrained optimisation. A population of answers is evolved until solutions that are good enough can be returned.

**Swarms** [1] proposes the usage of a swarm of micro-reasoners, each taking care of a part of a global set of entailment rules. The individuals of the swarm visit the graph and rewire it according to the knowledge they have.

### 3 Conclusion

The Semantic Web is a complex system made of large-scale, dynamic and potentially incoherent data. We showed how the typical tasks of Querying, Storage, Entailment, Consistency checking and Mapping can be rephrased from a logic problem into an optimization problem. This reformulation allows for considering evolutionary and swarm computing as a way to face the scaling and coherence challenges posed by the data from the Semantic Web.

### References

- [1] Kathrin Dentler, Christophe Guéret, and Stefan Schlobach. Semantic web reasoning by swarm intelligence. In *Proceedings of the 5th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2009) at ISWC*, October 2009.
- [2] Eyal Oren, Christophe Guéret, and Stefan Schlobach. Anytime query answering in RDF through evolutionary algorithms. In *ISWC*, volume 5318, pages 98–113. Springer Berlin Heidelberg, 2008.
- [3] Frank van Harmelen, Annette ten Teije, and Holger Wache. Knowledge engineering rediscovered: towards reasoning patterns for the semantic web. In *K-CAP*, pages 81–88, 2009.

# Text-Based Information Extraction Using Lexico-Semantic Patterns

Frederik Hogenboom      Wouter IJntema      Flavius Frasinca

*Erasmus University Rotterdam  
P.O. Box 1738, NL-3000 DR, Rotterdam, the Netherlands  
fhogenboom@ese.eur.nl, wouterijntema@gmail.com, frasinca@ese.eur.nl*

*The full version of this paper, entitled A Lexico-Semantic Pattern Language for Learning Ontology Instances from Text, will appear in Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Elsevier, 2012 (DOI: 10.1016/j.websem.2012.01.002)*

## Abstract

In order to cope with the ever increasing amount of data available on the Web, information extraction patterns are frequently employed to gather relevant information. Currently, most patterns use lexical and syntactic elements, but fail to exploit domain semantics. We propose a lexico-semantic pattern-based rule language, i.e., the Hermes Information Extraction Language (HIEL), which exploits a domain ontology for pattern creation. Experiments on financial news show that HIEL rules outperform lexico-syntactic rules and state-of-the-art lexico-semantic JAPE rules in terms of rule creation times and  $F_1$  scores.

## 1 Introduction

The tremendous growth of the Web has resulted in enormous amounts of data that are readily available to the average user. Many researchers have hence developed ways to convert these vast amounts of data into valuable information that can be used for various purposes, e.g., decision support or trading tools. For information extraction, patterns are frequently applied. For example, simple lexico-syntactic patterns [3] can be used to extract hyponyms from text. The problem with these type of patterns is that their support is often limited to hypernym, hyponym, meronym, and holonym relations. They employ limited syntactical elements and do not make use of the domain semantics. While these patterns generate high precision, recall lags behind, hence driving the development of lexico-semantic pattern languages like JAPE [1] to cope with this issue. However, these often suffer from verbosity and complexity, or semantic elements are not exploited to their full potential (e.g., by making use of a reasoning engine).

Therefore, we introduce the lexico-semantic pattern-based rule language called the Hermes Information Extraction Language (HIEL), which makes use of lexical and syntactical elements, as well as semantic elements. HIEL utilizes Semantic Web technologies by employing domain ontologies, hereby exploiting the domain concepts through the use of inference. Our language is evaluated in the Hermes news processing framework [2], of which the underlying ontology consists of lexicalized concepts for the financial domain.

## 2 Language Syntax

Figure 1 shows an example rule that links CEOs to their subjective companies, to illustrate the main features of our language. Lexical and syntactic elements are indicated by white labels, whereas semantic elements (which make use of the Hermes knowledge base) are indicated by shaded labels.

In HIEL, a rule typically consists of a left-hand side (LHS) and a right-hand side (RHS). Once the pattern on the RHS has been matched, it is used in the LHS, which consists of three components, i.e., a subject, predicate, and an object, where the predicate describes the relation between the subject and the object (in this case `hasCEO`). The RHS supports sequences of many different features, as explained below.

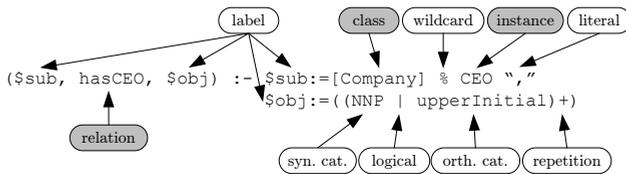


Figure 1: Example HIEL pattern

First, labels (preceded by  $\$$ ) on the RHS associate sequences using  $:=$  to the correct entities specified on the LHS. Second, syntactic categories (e.g., nouns, verbs, etc.) and orthographical categories (i.e., token capitalization) can be employed. Next, HIEL supports the basic logical operators *and* ( $\&$ ), *or* ( $|$ ), and *not* ( $!$ ), and additionally allows for repetition (regular expression operators, i.e.,  $*$ ,  $+$ ,  $?$ , and  $\{ \dots \}$ ). Moreover, wildcards are also supported, allowing for  $\geq 0$  tokens ( $\%$ ) or exactly 1 token ( $\_$ ) to be skipped.

Of paramount importance is the support for semantic elements through the use of concepts, i.e., ontological classes, which are defined as groups of individuals that share the same properties, i.e., the instances of a class. A concept may consist of alternative lexical representations that are stored using the synonym property. The hierarchical structure of the ontology allows the user to make rules either more specific or more generic, depending on the needs at hand.

### 3 Evaluation

We have implemented our language as an extension to the Hermes News Portal (HNP) [2] by adding a HIEL rule engine, rule editor, and annotation validator. In our experiments we use 500 financial news articles scraped from the Web and an ontology consisting of 65 classes, 18 object properties, 11 data properties, and 1,167 individuals. When comparing the creation times (in seconds) of domain experts for rule groups covering 10 different financial events, creating lexico-syntactic rules took 5,839 seconds until  $F_1$  scores reached 50%, whereas for HIEL rules it only took 356 seconds: a speedup of approximately 16 times. For JAPE rules, the creation times for the same task averaged to 806 seconds, which is about 2 times slower than HIEL. When allowing more creation time for HIEL and JAPE rules (up to 5,839 seconds), the  $F_1$  scores increase to 78.7% (83.9% precision, 74.1% recall) and 68.7% (85.3% precision, 57.5% recall), respectively, with respect to lexico-syntactic rules, which have an  $F_1$  score of 51.9% (54.9% precision and 49.3% recall). In all cases, HIEL rules outperform JAPE and lexico-syntactic rules in terms of  $F_1$  scores and creation times.

### 4 Conclusions

We have presented HIEL, a lexico-semantic pattern-based rule language which employs ontological elements (concepts) for information extraction. Through the addition of semantics, we obtain more generic, yet more accurate rules than their lexical counterparts. Additionally, the use of ontologies within the patterns promotes sharing of information as well as easy extendability. Initial experiments on a set of Web news articles show that in terms of creation times, HIEL rules outperform lexico-syntactic rules and the state-of-the-art JAPE rules. Moreover, within a fixed amount of time, HIEL rules yield higher  $F_1$  scores than the JAPE and lexico-syntactic rules.

### References

- [1] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 168–175. ACL, 2002.
- [2] Flavius Frasincar, Jethro Borsje, and Leonard Levering. A Semantic Web-Based Approach for Building Personalized News Services. *International Journal of E-Business Research*, 5(3):35–53, 2009.
- [3] Marti A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In *14th Conference on Computational Linguistics (COLING 1992)*, volume 2, pages 539–545. ACL, 1992.

# Doubtful Deviations and Farsighted Play

(Extended Abstract)

Wojciech Jamroga      Matthijs Melissen

*Computer Science and Communication, University of Luxembourg*

## 1 Introduction

Nash equilibrium (NE) defines stable play as one where, even if the players knew what the others are going to do, they would not deviate from their choices unilaterally. Conversely, if some player can beneficially deviate from strategy profile  $s$ , then the profile is assumed to describe irrational play. We point out that some of these deviations may not be profitable anymore if one takes into account the possibility of further deviations from the other players. As a remedy, we propose the concept of *farsighted pre-equilibrium* which takes into account only player  $i$ 's deviations that do not lead to decrease of  $i$ 's payoff even if some other deviations follow. In this abstract, we only sketch the idea. Readers interested in a more detailed exposition and technical results are referred to the original paper [3].

## 2 Summary of the Contribution

We propose a new solution concept that we call *farsighted pre-equilibrium*. The idea is to “broaden” Nash equilibrium in a way that does not discriminate solutions that look intuitively appealing but are ruled out by NE. Then, Nash equilibrium may be interpreted as a specification of play which is certainly rational, and strategy profiles that are *not* farsighted pre-equilibria can be considered certainly *irrational*. The area in between is the gray zone where solutions are either rational or not, depending on the detailed circumstances.

Our main motivation is predictive: we argue that a solution concept that makes too strong assumptions open up ways of possible vulnerability if the other agents do not behave in the predicted way. Nash equilibrium seems too restrictive in many games (Prisoner’s Dilemma being a prime example). We show that FPE does select non-NE strategy profiles that seem sensible, like the “all cooperate” strategy profile in the standard as well as the generalized version of Prisoner’s Dilemma. Moreover, we observe that FPE favors solutions with balanced distributions of payoffs, i.e., ones in which no player has significantly higher incentive to deviate than the others.

A natural way of interpreting deviations in strategy profiles is to view the deviations as moves in a “deviation game” played on the metalevel. We show that farsighted pre-equilibria in the original game correspond to subgame-perfect Nash equilibria in the meta-game. This is a strong indication that the concept that we propose is well rooted in game-theoretic tradition of reasoning about strategic choice.

Farsighted play has been investigated in multiple settings, starting from von Neumann and Morgenstern almost 70 years ago. Our proposal is (to our knowledge) the first truly noncooperative solution concept for farsighted play. In particular, it is obtained by reasoning about *individual* (meta-)strategies of *individually* rational players, rather than by reconstruction of the notion of *stable set* from coalitional game theory.

## 3 Farsighted Pre-Equilibria

Let  $G = (N, \Sigma_1, \dots, \Sigma_n, out_1, \dots, out_n)$  be a strategic game with  $N = \{1, \dots, n\}$  being a set of players,  $\Sigma_i$  a set of strategies of player  $i$ , and  $out_i : \Sigma \rightarrow \mathbb{R}$  the payoff function for player  $i$  where  $\Sigma = \Sigma_1 \times \dots \times \Sigma_n$  is the set of strategy profiles. We use the following notation:  $s_i$  is player  $i$ 's part of strategy profile  $s$ ,  $s_{-i}$  is the part of  $N \setminus \{i\}$ , and  $s \xrightarrow{i} s'$  denotes player  $i$ 's deviation from strategy profile  $s$  to  $s'$ .

**Definition 1.** Deviation  $s \xrightarrow{i} s'$  is locally rational iff  $out_i(s') > out_i(s)$ . Function  $F_i : \Sigma^+ \rightarrow \Sigma$  is a deviation strategy for player  $i$  iff for every finite sequence of profiles  $s^1, \dots, s^k$  we have that  $s^k \xrightarrow{i} F_i(s^1, \dots, s^k)$  is locally rational or  $F_i(s^1, \dots, s^k) = s^k$ . A sequence of locally rational deviations  $s^1 \rightarrow \dots \rightarrow s^k$  is  $F_i$ -compatible iff  $s^n \xrightarrow{i} s^{n+1}$  implies  $F_i(s^n) = s^{n+1}$  for every  $1 \leq n < k$ .

Locally rational deviations turn  $G$  into a graph in which the transition relation corresponds to Nash dominance in  $G$ . Deviation strategies specify how a player can (rationally) react to rational deviations done by other players.

**Definition 2** (Farsighted pre-equilibrium). Strategy profile  $s$  is a farsighted pre-equilibrium (FPE) if and only if there is no player  $i$  with a deviation strategy  $F_i$  such that: 1)  $out_i(F_i(s)) > out_i(s)$ , and 2) for every finite  $F_i$ -compatible sequence of locally rational deviations  $F_i(s) = s^1 \rightarrow \dots \rightarrow s^k$  we have  $out_i(F_i(s^1, \dots, s^k)) \geq out_i(s)$ .

As an example consider the following games (Prisoner’s Dilemma, Matching Pennies, and a variant of the latter with less symmetric payoffs):

(A)	<i>C</i>	<i>D</i>	(B)	<i>H</i>	<i>T</i>	(C)	<i>H</i>	<i>T</i>
<i>C</i>	(7, 7)	(0, 8)	<i>H</i>	(1, 0)	(0, 1)	<i>H</i>	(2, 1)	(0, 3)
<i>D</i>	(8, 0)	(1, 1)	<i>T</i>	(0, 1)	(1, 0)	<i>T</i>	(0, 1)	(1, 0)

It is easy to see that game (A) has two farsighted pre-equilibria:  $(C, C)$  and  $(D, D)$ , game (B) has no pre-equilibrium, and game (C) has one, namely  $(H, H)$ .

For more details and technical results, consult the original paper [3].

## 4 Related Work

The discussion on myopic vs. farsighted play dates back to von Neumann and Morgenstern’s *abstract stable set* in coalitional games [6], and Harsanyi’s *indirect dominance* of coalition structures [2]. A variety of farsighted solution concepts for coalitions were studied in further papers, e.g. [1, 4, 5]. The main difference between our farsighted pre-equilibrium and the other solution concepts discussed in this section lies in the perspective. It can be argued that the type of rationality defined in [6, 2, 1, 4, 5] is predominantly coalitional. This is because those proposals ascribe stability to *sets* of strategy profiles, which does not have a natural interpretation in the noncooperative setting. Moreover, some of the concepts are based on coalitional rather than individual deviations. In this sense, FPE is the first truly noncooperative solution concept for farsighted play that we are aware of.

## References

[1] M. Chwe. Farsighted coalitional stability. *Journal of Economic Theory*, 63:299–325, 1994.  
 [2] J. Harsanyi. Interpretation of stable sets and a proposed alternative definition. *Management Science*, 20:1472–1495, 1974.  
 [3] W. Jamroga and M. Melissen. Doubtful deviations and farsighted play. In *Proceedings of EPIA*, volume 7026 of *LNCIS*, 506–520, 2011.  
 [4] Suzuki, A., Muto, S.: Farsighted stability in an n-person prisoners dilemma. *International Journal of Game Theory* **33** (2005) 431–445  
 [5] N. Nakanishi. Purely noncooperative farsighted stable set in an n-player Prisoners Dilemma. Technical Report 707, Kobe University, 2007.  
 [6] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press: Princeton, NJ, 1944.

# Norm Contextualization <sup>1</sup>

Jie Jiang      Huib Aldewereld      Virginia Dignum      Yao-Hua Tan

*Delft University of Technology  
P.O.Box 5015 2600 GA, The Netherlands*

## 1 Introduction

This paper proposes a normative structure to represent and analyze sets of norms that takes into consideration both the interrelationships between different norms and the context of their applications. This extends current approaches where relations between norms are not explicitly considered. More importantly, the explicit representation of institutional contexts on norms facilitates a contextual refinement normative structure, i.e., a modular way of modeling norms through norm contextualization. Based on the mapping to Colored Petri Nets (CPNs), our normative structure will enable, given a set of norms in a specific context represented as a colored petri net, to check whether there is a possible way to comply with those norms, i.e., a path through the net which indicates norm compliant at all steps.

## 2 Normative structure

Our definition of *norm* is formalized by the ADIC syntax proposed by E. Ostrom [1], specifying who (Attribute) is obliged/forbidden/permitted (Deontic) to do or achieve what (alm), given what preconditions (Condition). To model the possible relations between the norms in a specific context, we introduce the concept of *norm net* in Definition 1.

**Definition 1** (Norm Net). A Norm Net  $NN = (context, NS)$ , where

- *context* is a set of states defined by predicates, and
- $NS = n$ , or  $NS = AND(NS_i, NS_j)$ , or  $NS = OR(NS_i, NS_j)$ , or  $NS = OE(NS_i, NS_j)$  where  $n$  is a norm,  $NS_i$ ,  $NS_j$ , and  $NS$  are norm sets.

Each norm net is associated with an institutional context determined by a set of states which concern but do not restrict to aspects such as individuality, activity, location, time, relation [2]. Contexts enable agents to control the evolution of a norm net and accommodate compliance and resolution of conflicts. A norm set  $NS$  is a nested structure composed of a set of hierarchically connected norms in a certain context. In a norm net, obligations and prohibitions may have corresponding sanctions while permissions usually do not. A norm and its sanctions are exclusive and conditional, i.e., either conform to the norm or accept the sanctions when violating the norm, which is in accordance with the semantic of *OE* (Or Else) operator.

For example, in the EU international trade regulations concerning the issue of *origin of goods*, a norm net can be constructed as  $NN_1 = (context_1, NS_1)$  where

- $context_1 =$  “origin of goods in the EU”,
- $NS_1 = AND(AND(n_1, n_2), OE(n_3, n_4))$ , where
  - $n_1$ : [role: Exporters] [deontic: should] [action: apply for certificate of origin] [condition: when exporting goods to the EU].
  - $n_2$ : [role: The customs authorities] [deontic: should] [action: issue certificate of origin to the qualified applicants].

---

<sup>1</sup>The full version of this paper has been accepted for publication at COIN@AAMAS2012.

- $n_3$ : [role: Importers] [deontic: must] [action: present Customs with a specific origin documents] [condition: at the moment of import].
- $n_4$ : [role: The customs authorities] [deontic: should] [action: reject the import] [condition: when the origin documents cannot be presented].

### 3 Norm Contextualization

Laws and regulations are a system of textual rules and guidelines that are enforced through social institutions to govern behavior. They are specified as a normative structure, which describes the expectations and boundaries for agent behavior. However, in real world domains, norms are not specified at a single level of abstraction. An abstract norm net, resulting from the formalization of law/regulation, may have different extensions according to different contexts. Usually, laws are first issued at a higher abstraction level stating the dos, don'ts and sanctions to regulate actors' behavior. Based on this set of abstract norms, elaboration will be conducted according to the specific characteristics and requirements of different situations, which results into sets of contextual norms. This elaboration process facilitates detailed explanation of abstract norms in a concrete implementing environment. Figure 1 shows an example of norm contextualization concerning the issue of *origin of goods*. From top to bottom, the norm refinement relation through contextualization reflects how norms are evolved in real life.

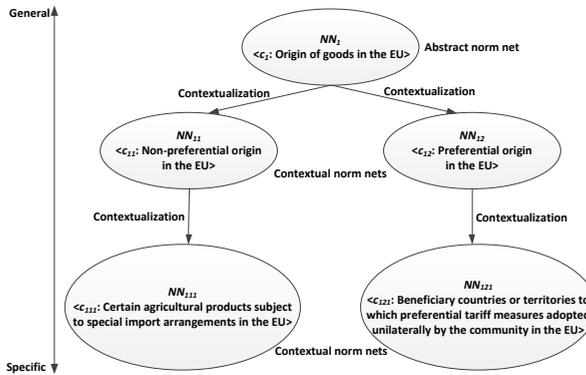


Figure 1: An example of norm contextualization

### 4 Verification

To enable consistency and compliance checking of norm nets, we introduce a verification based on the mapping to CPNs and make use of its behavioral properties. Roles in norms are mapped to the color sets in CPNs so that colored tokens correspond to role enacting agents in MAS. Actions in norms are mapped to the transitions in CPNs while conditions in norms are mapped to the guard functions in CPNs. Thus, only when the condition of a norm holds can the corresponding action specified in the norm be permitted, obliged or forbidden. Places in CPNs indicate the states of the role enacting agents, i.e., their behavior status in terms of norms. For the three deontic representations in norms and the three relation operators in norm nets, we use different building blocks with special places and transitions.

### References

[1] E. Ostrom. *Understanding institutional diversity*. Princeton University Press, 2005.

[2] A. Zimmermann, A. Lorenz, and R. Oppermann. An operational definition of context. In *Modeling and Using Context*, LNCS, 2007.

# Automated Measurement of Spontaneous Surprise<sup>1</sup>

Bart Joosten <sup>a</sup>      Eric Postma <sup>a</sup>      Emiel Krahmer <sup>a</sup>      Marc Swerts <sup>a</sup>

Jeesun Kim <sup>b</sup>

<sup>a</sup> *TiCC, Tilburg University, Tilburg, The Netherlands*

<sup>b</sup> *MARCS Auditory Laboratories, University of Western Sydney, Sydney, New South Wales 1797, Australia*

## 1 Introduction

This paper presents the Spontaneous Surprise Measurement (SSM) method, an automatic method for the measurement of the facial expressions associated with spontaneous surprise as opposed to acted surprise. Prototypical or acted surprise is typically associated with three clearly distinctive facial displays, viz., raising of the eyebrows, widening of the eyes, and opening of the mouth. However, spontaneous surprise is characterised by a more subtle and person-specific combination of the aforementioned displays. Automatically detecting spontaneous surprise could greatly benefit automatic tutoring or dialog systems.

We have adopted an approach described in [3] to perform computational analysis on video data acquired from carefully designed experiments to measure facial expressions. We used a novel experimental paradigm developed by two of the authors (MS and JK) to elicit and label spontaneous surprise. In this experimental setup participants were lead to believe they were participating in a memory task. They were presented with a cover story that we were interested in how memory is affected by context and by reading aloud words. In the first stage of the experiment the participants were asked to imagine words that fit a specific context (e.g. organs of the human body in the neutral condition versus favorite food items among Dutch children in the surprise condition). In the second stage the participants were asked to read aloud each of a sequence of 10 words (of either of the two contexts) displayed on a screen. In both conditions, the target word liver is one of displayed words. We video recorded 27 Dutch participants using a hidden camera that is positioned behind the computer screen so that facial expressions can be clearly captured. In the third stage the participants were asked to recall as many words as possible. Crucially, we elicit the target word (liver) in two conditions: a neutral condition in which the word clearly fits in the context organs of the human body and a surprise condition in which the word is highly unexpected as a favorite food item among Dutch children.

In most participants, the paradigm results in clearly different behavior in both experimental conditions (neutral versus surprise). Careful visual inspection of the video fragments revealed five distinctive facial expressions in the surprise condition, viz. (1) eyebrow frowning, (2) eyebrow raising, (3) widening eyes (4) mouth opening, and (5) brief head retraction. The presence and prominence of each of the expressions differed from participant to participant, but eyebrow frowning (1) was the most prevalent expression overall. Therefore, we manually annotated the frames for all participants who displayed the eyebrow frown (11 in total).

## 2 Spontaneous Surprise Measurement method

According to [2], surprise is prototypically displayed by three facial actions: raising of the eyebrows, opening of the mouth, and widening of the eyes. The Spontaneous Surprise Measurement (SSD) method aims

---

<sup>1</sup>The full version of this paper is accepted at *Measuring Behavior*, August 2012, Utrecht

at measuring surprise by focusing on the first of the three actions. To that end, the SSD method consists of three steps: the identification of landmarks, the texture analysis of the eyebrow region, and the classification of surprise. The landmark-identification step is realized by using the Constrained Local Model [4]. Given a video frame, it returns the locations of a number of predefined facial landmarks, such as, the nose tip, corners of the eyes, and the eyebrows. The texture analysis is restricted to an image patch covering the facial region of the eyebrows and is performed using a multi-scale Gabor filter-bank [1]. For each image position (pixel), the Gabor filter-bank returns  $N \times M$  energy values representing the presence of oriented visual structure of a certain thickness (spatial frequency), where  $N$  represents the number of orientations and  $M$  the number of spatial frequencies. Finally, the classification maps the (aggregated) energy values onto the binary classes frown and non-frown and subsequently to surprise and no surprise. We used a leave-one-subject-out scheme to train and test a support vector machine (SVM) classifier on the frames annotated for the presence or absence of frowns. If in a video sequence our classifier would classify three consecutive frames as frowns, we classify that sequence as surprise.

### 3 Results and Conclusions

An impression of the current leaving-one-subject-out performances of the SSM method as determined with an SVM classifier is given in tables 1(a) and 1(b). Both tables contain confusion tables listing the frown-detection and surprise-detection results respectively. Overall, the confusion tables show that non-frowns are often detected as frowns and that neutral sequences are often detected as surprise sequences. The higher scores on the main diagonal indicate that the SSM method is reasonably successful in detecting spontaneous surprise.

Table 1: Confusion matrices classification results

(a) Frown classification			(b) Surprise classification		
Actual	Predicted		Actual	Predicted	
	frown	non-frown		surprise	neutral
frown	133	72	surprise	19	10
non-frown	1666	3677	neutral	4	13

We have developed a method for the measurement of Spontaneous Surprise that is reasonably successful at detecting frowning as part of one of the key facial actions for surprise (raising the eyebrows). Further optimizing the texture-analysis and classification stages may lead to an improvement in performance. However, our aim is to yield improvement by incorporating domain knowledge about the facial features signaling surprise. More specifically, we want to extend the SSM method by including measurements of the two other facial actions proposed by Ekman, i.e., opening of the mouth, and widening of the eye. In addition, mouth opening and brief head retraction will also be examined.

### References

- [1] I.J. Bereznoy, E.O. Postma, and H.J. van den Herik. Computer analysis of van goghs complementary colours. *Pattern Recognition Letters*, 28(6):703–709, 2007.
- [2] P. Ekman. Universal facial expressions of emotion. In U. Segerstrale and P. Molnar, editors, *Where Nature Meets Culture*, pages 27–46, 1997.
- [3] B. Joosten, M.A.A. van Amelsvoort, E.J. Krahmer, and E.O. Postma. Thin slices of head movements during problem solving reveal level of difficulty. In G. Salvi, J. Beskow, O. Engwall, and S. Al Moubayed, editors, *Proceedings of the International Conference on Audio-Visual Speech Processing (AVSP 2011)*, pages 85–88, 2011.
- [4] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

# Why Won't You Do What's Good for You? Using Intelligent Support for Behavior Change

Michel Klein <sup>a</sup>

Nataliya Mogles <sup>a</sup>

Arlette van Wissen <sup>a</sup>

<sup>a</sup> *Department of Artificial Intelligence, VU University Amsterdam  
De Boelelaan 1081, 1081 HV Amsterdam*

## 1 Introduction

A good health requires a healthy lifestyle. It is however challenging to find (and keep) the optimal balance between work, a social life and, for example, a healthy diet or medicine schedule. People with a chronic disease face additional challenges, such as physical discomforts and side-effects of medicine intake. Adopting a healthy lifestyle therefore often requires behavior change. In short, people have lots of reasons not to do what's good for them. Consequently, the number of people that have obesity or a chronic disease such as diabetes type 2 has increased considerably over the past years [4].

The use of computers to support patients with their self-management (i.e., engagement and empowerment with respect to their health condition) has proven to be an effective approach to improve therapy adherence [5]. These therapies can consist of lifestyle advice and/or precise instructions for medication intake. Although intelligent persuasive assistants are increasing in popularity for the use of behavior interventions, those assistants are rarely based on formal models of behavior change [2]. Yet in order to design an effective support system, it is necessary to take a closer look at the underlying mechanisms of behavior change and how they can be influenced to establish the desired behavior.

This work [1] addresses this issue and presents a computational model based on theoretical frameworks of behavior change. The model is developed to function as the core of a reasoning mechanism of an intelligent support system – called eMate – that is able to create theory-based intervention messages that stimulate a healthy lifestyle. eMate first tries to determine the cause of unhealthy behavior by asking short questions via a mobile phone application and by gathering information from an online lifestyle diary. The system then attempts to influence and improve the user's behavior using tailored information and persuasive motivational messages.

## 2 A Model of Behavior Change: COMBI

For health interventions to be effective, they need to incorporate existing theories on behavior change and persuasive design. The model of behavior change designed in this work is based on several existing models from psychology literature that describe determinants for behavior change. The Transtheoretical Model [3] forms the basis for the proposed model, identifying five stages of behavior change: *precontemplation*, *contemplation*, *preparation*, *action*, and *maintenance*. The model denotes causal dependencies which represent transitions from one state to another. These transitions occur if the value of a state exceeds a certain threshold. The COMBI model further differentiates between internal and external determinants of behavior, incorporating key components from existing psychological theories (see Figure 1). The integrated model can be used to analyze how the behavioral determinants influence each other and how they can be manipulated to influence behavior. Some simulations have been performed to study the interplay between the different determinants of the model. These simulations show that the model can account for behavioral phenomena found in psychology and sociology.

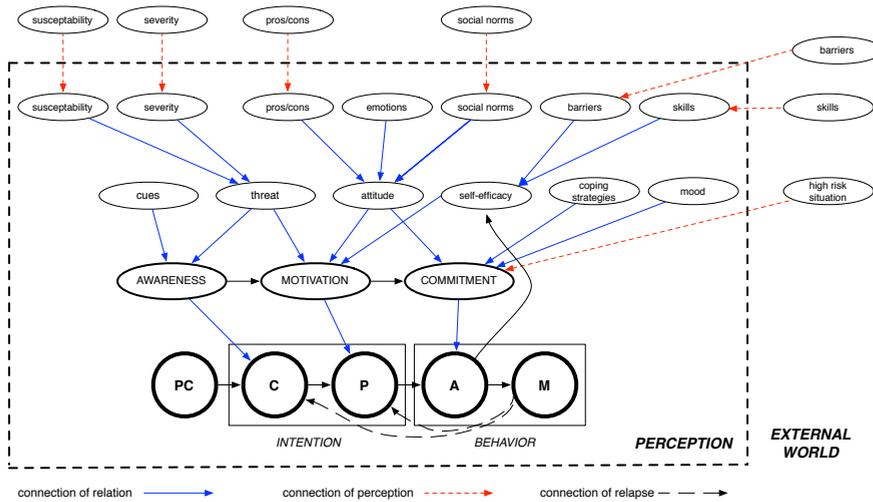


Figure 1: The integrated model of behavior change COMBI

### 3 Intelligent Support System: eMate

The COMBI model has been used as basis for an intelligent coaching system, called eMate. The eMate system aims to support patients with Diabetes Mellitus type II, HIV or cardiovascular disease in adhering to their therapy. The model is used to analyze the state of the patient with respect to his or her behavior change goals. This is done by investigating on a regular basis which of the factors that influence behavior change are probably the most problematic for the patient at that time. A rule-based representation of the model is implemented to allow for backward reasoning over the psychological factors in the model. Using this model-based diagnoses, the system deduces for which factors the value should be determined. For these identified factors, specific questions from psychological surveys are posed to the user on the mobile phone. The answers to these questions translate to values for these factors. After this diagnostic phase, the system determines which factor should be targeted to support the user in the most effective way. Every week, the user will receive a summary of his or her behavior and a motivating message related to the relevant factor via the website and the mobile phone.

Concluding, the integrated COMBI model is an example of a causal modeling approach to develop complex, user-tailored interventions aimed at behavior change. The model can be incorporated in a coaching system, which has a strong potential of providing support for individuals with respect to their lifestyles. Additionally, the system can be a diverse and useful tool for researchers to investigate how people can be motivated to adhere to their lifestyle goals.

### References

- [1] Michel C.A. Klein, Nataliya Mogles, and Arlette van Wissen. Why won't you do what's good for you? using intelligent support for behavior change. In Albert Salah and Bruno Lepri, editors, *Human Behavior Understanding*, volume 7065 of *LNCS*, pages 104–115. Springer Berlin / Heidelberg, 2011.
- [2] S. Michie, M. Johnston, J. Francis, W. Hardeman, and M. Eccles. From theory to intervention: mapping theoretically derived behavioural determinants to behaviour change techniques. *Applied Psychology*, 57(4), 2008.
- [3] J.O. Prochaska and C.C. DiClemente. *The Transtheoretical Approach: Crossing the Traditional Boundaries of change*. J. Irwin, Homewood, IL, 1984.
- [4] J.E. Shaw, R.A. Sicree, and P.Z. Zimmet. Global estimates of the prevalence of diabetes for 2010 and 2030. *Diabetes Research and Clinical Practice*, 87(1):4 – 14, 2010.
- [5] H. de Vries and J. Brug. Computer-tailored interventions motivating people to adopt health promoting behaviours: introduction to a new approach. *Patient Educational Counseling*, 36:99–105, 1999.

# Moment Constrained Semi-Supervised LDA

Marco Loog

*Pattern Recognition Laboratory, Delft University of Technology, The Netherlands*  
*The Image Group, University of Copenhagen, Denmark*

## Abstract

This *BNAIC compressed contribution* provides a summary of the work originally presented at the First IAPR Workshop on Partially Supervised Learning and published in [5]. It outlines the idea behind supervised and semi-supervised learning and highlights the major shortcoming of many current methods. Having identified the principal reason for their limitations, it briefly sketches a conceptually different take on the matter for linear discriminant analysis (LDA). Finally, the contribution hints at some of the results obtained. For any details, the reader is of course referred to [5].

## 1 Semi-Supervision and Current Limitations

Supervised learning aims to learn from examples. That is, given a limited number of instances of a particular input-output relation, its goal is to generalize this relationship to new and unseen data in order to enable the prediction of the associated output given new input. Specifically, supervised classification aims to infer an unknown feature vector-class label relation from a finite, potentially small, number of input feature vectors and their associated, desired output class labels. Now, an elementary question in pattern recognition and machine learning is whether and, if so, how the availability of additional unlabeled data can significantly improve the training of such classifier. This is what constitutes the problem of semi-supervised classification or, generally, semi-supervised learning [2].

The hope or, rather, belief is that semi-supervision can bring enormous progress to many scientific and application areas in which classification problems play a key role, simply by exploiting the often enormous amounts of unlabeled data available (think computer vision, text mining, retrieval, medical diagnostics, but also social sciences, psychometrics, econometrics, etc.). The matter of the fact, however, is that up to now semi-supervised methods have not been widely accepted outside of the realms of computer science, being little used in other domains. Part of the reason for this may be that current methods offer no performance guarantees [1] and often deteriorate in the light of large amounts of unlabeled samples [2, Chapter 4].

## 2 Sketch of a Different Take

In line with [4], [5] identifies as main reason for the frequent failure of semi-supervision that current semi-supervised approaches typically rely on assumptions *extraneous* to the classifier being considered. If, however, these *additional* assumptions are not accurate, such approach may obviously fail.

Focusing on classical LDA, the approach from [5] instead exploits the fact that the parameters that are to be estimated, i.e., class means  $m_i$  and a within covariance matrix  $\mathbf{W}$ , fulfill particular *intrinsic* relations. In particular, there are two relations that link label-dependent with label-independent quantities: one links the  $K$  class means  $m_i \in \mathbb{R}^d$  to the overall data mean  $\mu \in \mathbb{R}^d$  through  $\sum_{i=1}^K p_i m_i = \mu$  (with  $p_i$  the class priors) and the other links the between and within covariance with the total covariance:  $\mathbf{B} + \mathbf{W} = \mathbf{\Theta}$  [3]. In this way, class-independent parameters like  $\mu$  and  $\mathbf{\Theta}$ , which can be more accurately estimated using the additional unlabeled data, impose constraints on the parameters relevant to LDA,  $m_i$  and  $\mathbf{W}$ , leading to a reduction in

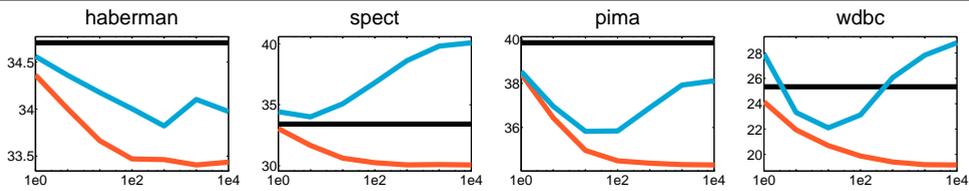


Figure 1: Mean error rates (vertical axis, averages over 1,000 repetitions) for the supervised (black), the proposed constrained semi-supervised (orange), and the self-learned classifier (light blue) on four real-world UCI Machine Learning Repository (Asuncion and Newman, 2007) data sets for various unlabeled sample sizes (horizontal axis, logarithmic scale) and a total of ten labeled training samples.

variability of these label dependent estimates. As a result, the performance of this semi-supervised linear discriminant is expected to improve over that of its supervised equal and typically does not deteriorate with increasing numbers of unlabeled data.

The ad hoc approach employed in [5] to find parameters that indeed satisfy the two above constraints is as follows<sup>1</sup>. Simply transform all labeled feature data  $x$  into  $x' = \Theta^{\frac{1}{2}} \mathbf{T}^{-\frac{1}{2}}(x - m) + \mu$  with  $\mathbf{T}$  the total covariance and  $m$  the total mean over all *labeled* data and with  $\Theta$  and  $\mu$  their counterparts as determined on *all* data, labeled as well as unlabeled. The label dependent statistics determined on  $x'$  now fulfill the necessary constraints. Note that these constraints are already automatically fulfilled in the supervised setting as in case  $\mu = m$  and  $\Theta = \mathbf{T}$ . The constraints only come into effect when additional unlabeled data is used.

### 3 Impression of Experimental Results

The few experimental results displayed in Figure 1 give an impression of the potential behavior of the constrained semi-supervised approach (in orange) in comparison to the standard, supervised setting (in light blue) and LDA trained by means of a common semi-supervised method (in black) typically referred to as self-training or self-learning [2, Chapter 1]. Results similar to those obtained by self-learning would have been obtained by the classical EM approach [2, Chapter 3]. In these experiments the proposed constrained approach improves over the supervised and self-learned approach in all cases. Additional results, examples where also this new approach may still fail to improve upon the supervised setting, and some further limitations are discussed in the original contribution [5].

## References

- [1] S. Ben-David, T. Lu, and D Pál. Does unlabeled data provably help? Worst-case analysis of the sample complexity of semi-supervised learning. In *COLT 2008*, pages 33–44, 2008.
- [2] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.
- [3] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
- [4] M. Loog. Constrained parameter estimation for semi-supervised learning: The case of the nearest mean classifier. In *ECML PKDD*, volume 6322 of *LNAI*, pages 291–304. Springer, 2010.
- [5] M. Loog. Semi-supervised linear discriminant analysis using moment constraints. In *Partially Supervised Learning*, volume 7081 of *LNAI*, pages 32–41. Springer, 2012.
- [6] M. Loog and A.C. Jensen. Constrained log-likelihood-based semi-supervised linear discriminant analysis. In *S+SSPR*, volume 7626 of *LNCS*. Springer, 2012.

<sup>1</sup>A more principled approach can be found in [6].

# Elitist Archiving for Multi-Objective Evolutionary Algorithms: To Adapt or Not To Adapt

Hoang N. Luong

Peter A. N. Bosman

*Centrum Wiskunde & Informatica, Amsterdam, The Netherlands  
Hoang.Luong@cwi.nl, Peter.Bosman@cwi.nl*

---

The full paper has been accepted for presentation at the *12th International Conference on Parallel Problem Solving From Nature*.

---

## 1 Introduction

Multi-objective optimization (MOO) problems involve solving multiple conflicting objectives at the same time. Each solution represents a compromise between different objectives, and a *utopian* solution optimizing all objectives is thus unattainable. The complete answer for an MOO problem is the Pareto-optimal set  $\mathcal{P}_S$  of all *non-dominated* trade-off solutions in the parameter space and its corresponding image  $\mathcal{P}_F$  in the objective space. The actual size of  $\mathcal{P}_S$  and  $\mathcal{P}_F$  can be infinite or too numerous to be obtained by finite computational resources. In practice, the desired result is often a diverse and representative subset  $\mathcal{S}$  of  $\mathcal{P}_S$  having a reasonable size with its image  $f(\mathcal{S})$  well-spread along the Pareto-optimal front  $\mathcal{P}_F$ .

Evolutionary algorithms (EAs) have long been considered as well-suited for solving MOO problems. Along with traditional EA components and operators, multi-objective evolutionary algorithms (MOEAs) often use a separate data structure, called the elitist archive, to keep track of the best Pareto set of non-dominated solutions. Archiving strategies are thus needed to maintain the archive around reasonable sizes while ensuring its diversity and proximity regarding the Pareto-optimal front. This paper presents and compares two elitist archiving strategies: a rigid grid discretization and its new adaptive version.

## 2 Rigid Grid Discretization (RGD)

Objective-space discretization is a popular method to control the elitist archive size for evolutionary multi-objective optimization and avoid problems with convergence. Each objective dimension  $f_i$  is divided into equal segments of unit length  $\lambda_i$  (in this paper,  $\forall i, \lambda_i = \lambda$ ). Overall, the objective space is discretized into equal hypercubes, and each hypercube is allowed to contain only one solution at a time. Because the edge-lengths  $\lambda_i$  of hypercubes are fixed during an MOEA run, this method is termed rigid-grid discretization.

A newly generated non-dominated solution competes with other solutions to enter the elitist archive. If the new solution is (Pareto) dominated by any archive solutions, it is discarded. A new non-dominated solution can enter the elitist archive if and only if it occupies an empty hypercube or it dominates the solution that currently resides in the same hypercube. If the new non-dominated solution does not dominate the occupant, that new solution is considered as a dominated solution and is discarded as well. When a new solution is accepted into the archive, all solutions dominated by it are removed from the archive.

## 3 Adaptive Grid Discretization (AGD)

Instead of setting edge-lengths  $\lambda_i$  of hypercubes, the practitioner decides the allowable archive size  $t$  regarding available resources. The adaptive archive functions much like in the rigid case: non-dominated solutions enter the archive, and dominated solutions are removed. When the archive size deviates too much from the target size  $t$ , the edge-lengths need to be re-determined. For reasons of computational efficiency, we set an upper bound  $t^{up}$  and a lower bound  $t^{low}$  for the elitist archive size. As soon as the archive size reaches the upper bound, the objective space adaptation process is triggered. AGD first determines the ranges of all current archived solutions in the objective space, and then performs a binary search, targeted at  $t^{low}$ , for how many segments each range should be divided into. The final discretization must satisfy the condition

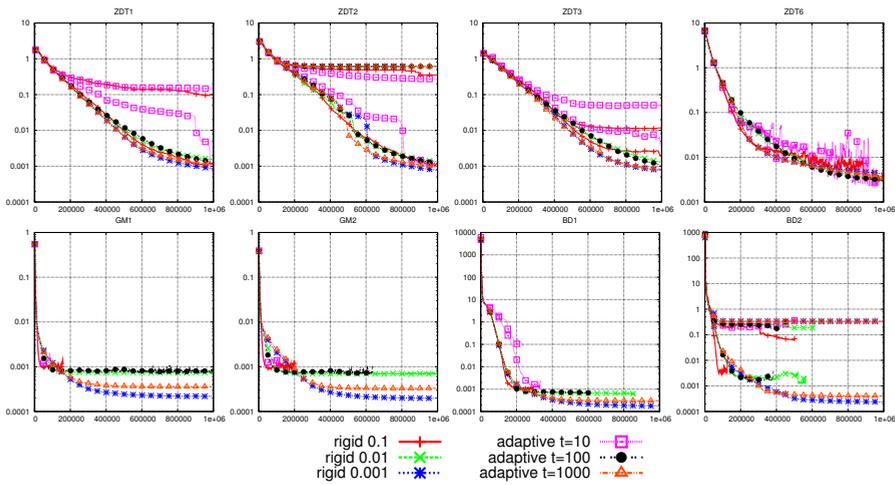


Figure 1: Horizontal axis: number of evaluations. Vertical axis:  $D_{\mathcal{P}_F \rightarrow \mathcal{S}}$ . Averages are shown both for successful runs and unsuccessful runs, giving double occurrences of lines if some runs were unsuccessful.

$t^{low} < t < t^{up}$ . In this paper, we set  $t^{low}$  and  $t^{up}$  as  $0.75 * t$  and  $1.25 * t$ , respectively. AGD can be seen as a sequence of RGDs with different discretization levels  $\lambda$ . There is an iteration  $g$  when solutions in the elitist archive already cover the ranges of the Pareto-optimal front and the maximal number of non-dominated solutions that the current grid can contain is close to the target size  $t$ . From that iteration  $g$ , there is no need to re-discretize the objective space any more.

### 4 Summarized Results and Conclusions

This paper uses the MAMaLGaM (Multi-objective Adapted Maximum-Likelihood Gaussian Model [1]) as the MOEA to be combined with the two elitist archiving strategies. Fig. 1 shows convergence graphs of the inverse generational distance indicator  $D_{\mathcal{P}_F \rightarrow \mathcal{S}}$  from the beginning until termination for MAMaLGaM on eight benchmark problems. The formulations of  $D_{\mathcal{P}_F \rightarrow \mathcal{S}}$  and benchmark problems can be found in the full paper. When the elitist archive has limited volume (i.e., the target size is too small,  $t = 10$ , or the grid is too coarse-grained,  $\lambda = 0.1$ ), it is less likely to achieve the desirable convergence ( $D_{\mathcal{P}_F \rightarrow \mathcal{S}} \leq 0.01$ ). Otherwise, when having archives of adequate capacity, the MOEA achieves good convergence behavior for both variants of archiving mechanisms. Fig. 1 also shows that the greater the elitist archive is, the better  $D_{\mathcal{P}_F \rightarrow \mathcal{S}}$  indicator values it can obtain. Because of allowing more solutions in the elitist archive, The  $D_{\mathcal{P}_F \rightarrow \mathcal{S}}$  indicator values of RGD are thus slightly better than those of the corresponding AGD (i.e.,  $\lambda = 0.1$  vs  $t = 10$ ,  $\lambda = 0.01$  vs  $t = 100$ ,  $\lambda = 0.001$  vs  $t = 1000$ ). Doubling  $t$  would give similar  $D_{\mathcal{P}_F \rightarrow \mathcal{S}}$  indicator values. Also, if we terminate an MOEA run when it reaches the successful threshold ( $D_{\mathcal{P}_F \rightarrow \mathcal{S}} \leq 0.01$ ), the adaptive and rigid archives have similar convergence behavior.

The two variants of objective-space discretization archiving strategies are showed to have similar convergence behavior. Interested readers are invited to read the full paper for detailed comparison on the basis of front quality, success rate, and running time. The advantage of the adaptive archiving strategy resides in its straightforwardness and transparency through which the practitioners can decide their desirable archive size and all the archiving processes are then automatically handled. Our technique is able to select appropriate discretizations such that the final approximation set is well-spread with good proximity concerning the Pareto-optimal front provided that the MOEA is capable of generating such good solutions. Experimental results support our above claims.

### References

[1] Peter A. N. Bosman: The Anticipated Mean Shift and Cluster Registration in Mixture-based EDAs for Multi-Objective Optimization. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation - GECCO '10, ACM New York, NY, USA (2010) 351–358

# AU Classification using AAMs and CRFs

Laurens van der Maaten <sup>a</sup>

Emile Hendriks <sup>a</sup>

<sup>a</sup> *Delft University of Technology, Mekelweg 4, 2628 CD Delft*

Automatic facial expression recognition is an important problem in social signal processing that has applications ranging from treatment of autistic children to monitoring of conflict situations [6]. In psychology, facial expressions are generally described using the Facial Action Coding System (FACS; [2]), in which each facial muscle is referred to as an action unit (AU) that is present (i.e. muscle contracted) or not present (i.e. muscle relaxed). We developed a system that automatically classifies AUs based on (variations in) facial texture and shape features. The feature extraction is performed with the help of an active appearance model [1]. Detection of AU presence is performed by training a newly developed structured prediction algorithm [5] on the features thus obtained. A complete description of our system was published in [4].

**Facial feature point detection.** Active appearance models (AAMs) are deformable template models that consist of two main parts: (1) a facial shape model and (2) a facial texture model. The shape model is obtained by performing PCA on a collection of manual feature-point annotations that are Procrustes-aligned, i.e. by fitting a linear-Gaussian latent variable model to normalized feature points. The resulting model can be used to generate likely configurations of facial feature points. The texture model is obtained by performing PCA on a collection of shape-normalized texture images, i.e. by using the feature point annotations to warp the annotated face images to a single coordinate frame and fitting a linear-Gaussian latent variable model to the resulting normalized face images. The texture model can be used to generate likely facial appearances that are normalized for shape variations in the face. AAMs combine the shape and texture model by warping the generated facial texture onto the generated facial shape. Identification of facial feature points using AAMs is performed by maximizing the likelihood of the observed face image under the AAM with respect to its latent variables, i.e. by minimizing the squared error between the observed face image and the facial appearance generated by the AAM.

**Feature extraction.** Using the identified facial feature points, we extract two main types of features: (1) normalized shape variations and (2) shape-normalized texture variations. Normalized shape variations (NSV) measure the difference between feature point locations in the current frame and feature point locations in the first frame of each movie. NSV features are computed by Procrustes-aligning the identified face shape to the base shape to remove rigid transformations and, subsequently, subtracting the resulting coordinates from the coordinates in the first frame. Whilst shape variations are important in AU detection predicting the presence of, e.g., action unit 24 (lip pressor) requires textural information on wrinkles. Shape-normalized texture variations (SNTVs) capture such texture information by warping each face image onto the base shape (see Figure 3), and subtracting the resulting image from the the shape-normalized image in the first frame.

**Action unit detection.** The detection of AU presence based on the extracted facial shape and texture variation features is performed via an extension of the conditional random field (CRF) model called hidden-unit CRF [5]. Hidden-unit CRFs model latent structure in the data that is relevant for classification in

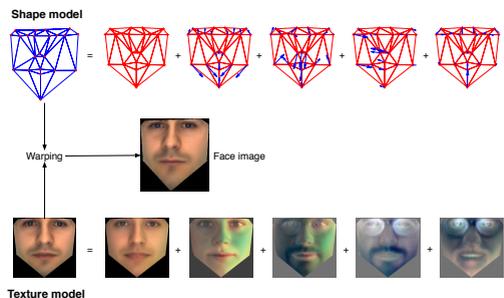


Figure 1: Active appearance modeling: (1) the face shape is made by adding a linear combination of shape components to the base shape, (2) the facial texture is made by adding a linear combination of the texture components to the mean texture, and (3) the final face image is made by warping the texture onto the shape.

a collection of binary stochastic variables that are conditionally independent given the data and the label sequence (see Figure 2). This conditional independence property facilitates efficient inference and learning. The resulting model can represent much more complex decision boundaries than standard CRFs.

**Experiments.** We performed experiments on the Cohn-Kanade data set, which comprises 593 short movies of 123 subjects performing a single posed expression. The movies are labeled for the AUs present in the movie. We measure generalization errors of classifiers trained to predict the presence of single AU in terms of the area under the ROC curve (AUC) via leave-one-subject-out cross-validation (see Table 4) on both feature sets. The results show that both types of features provide a lot of information on the presence of AUs: with these performances, our system will likely pass official FACS-certification exams. For AUs that cause large movements of feature points (like AU1, AU2, and AU25), shape variations (NSV) are the most informative features, whereas texture variations (SNTV) are most informative in recognizing the other AUs.

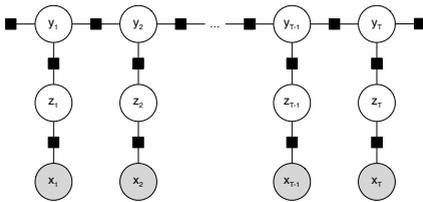


Figure 2: Factor graph of hidden-unit CRF:  $x$  denotes data units,  $z$  denotes binary stochastic hidden units, and  $y$  denotes 1-of- $K$  label units.

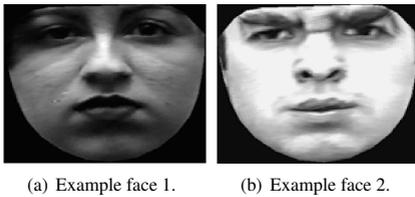


Figure 3: Shape-normalized face textures: feature points are in the same location.

AU	Name	NSV	SNTV
<b>1</b>	<i>Inner Brow Raiser</i>	<b>0.8947</b>	0.8834
<b>2</b>	<i>Outer Brow Raiser</i>	<b>0.9278</b>	0.9270
<b>4</b>	<i>Brow Lowerer</i>	0.8277	<b>0.8667</b>
<b>5</b>	<i>Upper Lip Raiser</i>	0.8857	<b>0.9070</b>
<b>6</b>	<i>Cheek Raiser</i>	<b>0.8740</b>	0.8691
<b>7</b>	<i>Lip Tightener</i>	0.8484	<b>0.8633</b>
<b>9</b>	<i>Nose Wrinkler</i>	<b>0.9415</b>	0.9401
<b>11</b>	<i>Nasolabial Deep.</i>	0.8818	<b>0.9270</b>
<b>12</b>	<i>Lip Corner Puller</i>	0.9171	<b>0.9222</b>
<b>15</b>	<i>Lip Corner Depr.</i>	0.9178	<b>0.9239</b>
<b>17</b>	<i>Lower Lip Depr.</i>	0.9017	<b>0.9125</b>
<b>20</b>	<i>Lip Stretcher</i>	0.8713	<b>0.8918</b>
<b>23</b>	<i>Lip Tightener</i>	0.9399	<b>0.9412</b>
<b>24</b>	<i>Lip Pressor</i>	0.9275	<b>0.9408</b>
<b>25</b>	<i>Lips Part</i>	<b>0.9075</b>	0.8961
<b>26</b>	<i>Jaw Drop</i>	0.8847	<b>0.8876</b>
<b>27</b>	<i>Mouth Stretch</i>	0.9455	<b>0.9459</b>
<b>ALL</b>	<i>Averaged</i>	0.8997	<b>0.9086</b>

Figure 4: Average AUC of hidden-unit CRFs on both features. Best performance is boldfaced.

**Future work.** An important issue of the current system is that its performance is not robust under out-of-plane rotations or partial occlusions of the face. We did not test our system in such situations, because to the best of our knowledge, there are no publicly available databases that contain FACS-coded videos with out-of-plane rotations and/or occlusions. In future work, we aim to extend our CRF models to exploit correlations between action units [3]: for instance, if we detect the presence of action unit AU12, the probability that AU13 or AU14 are also present increases; our models should exploit this information. We also plan to use our system to detect basic emotions and higher-level social signals, such as agreement and disagreement, by learning mappings from action unit labels to these emotions or signals.

## References

- [1] T.F. Cootes, G. Edwards, and C.J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 484–498, 1998.
- [2] P. Ekman and E. Rosenberg. *What the face reveals (2nd edition)*. Oxford, New York, NY, 2005.
- [3] C. Sutton, A. McCallum, and K. Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research*, 8(Mar):693–723, 2007.
- [4] L.J.P. van der Maaten and E.A. Hendriks. Action unit classification using active appearance models and conditional random fields. *Cognitive Processing*, 2012.
- [5] L.J.P. van der Maaten, M. Welling, and L.K. Saul. Hidden-unit conditional random fields. *JMLR W&CP*, 15:479–488, 2011.
- [6] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27:1743–1759, 2009.

# Playout Search for Monte-Carlo Tree Search in Multi-Player Games <sup>1</sup>

J. (Pim) A. M. Nijssen

Mark H.M. Winands

Department of Knowledge Engineering, Faculty of Humanities and Sciences  
Maastricht University, Maastricht, The Netherlands

Over the past years, Monte-Carlo Tree Search (MCTS) [2, 3] has become a popular technique for playing deterministic perfect-information multi-player games. MCTS is a best-first search technique that instead of an evaluation function uses simulations to guide the search. For MCTS, a tradeoff between search and knowledge has to be made. The more knowledge is added, the slower each playout gets. The trend seems to favor fast simulations with computationally light knowledge, although recently, adding more heuristic knowledge at the cost of slowing down the playouts has proven beneficial in some games [8].

In this abstract we propose Playout Search for MCTS in multi-player games. Instead of playing random moves biased by computationally light knowledge in the playout phase, domain knowledge can be incorporated by performing small searches. These searches employ more expensive evaluation functions to assess the leaf nodes of non-terminal positions. This reduces the number of playouts per second significantly, but it improves the reliability of the playouts. When selecting a move in the playout phase, one of the following three search techniques is used to choose a move.

**Two-ply max<sup>n</sup>** [4]. A two-ply max<sup>n</sup> search tree is built where the current player is the root player and the first opponent plays at the second ply. Both the root player and the first opponent try to maximize their own score.  $\alpha\beta$ -pruning in a two-ply max<sup>n</sup> search tree is not possible.

**Two-ply Paranoid** [7]. Similar to max<sup>n</sup>, a two-ply search tree is built where the current player is the root player and the first opponent plays at the second ply. The root player tries to maximize its own score, while the first opponent tries to minimize the root player's score. Contrary to max<sup>n</sup>,  $\alpha\beta$ -pruning is possible.

**Two-ply Best Reply Search (BRS)** [5]. The tree structure of BRS is similar to that of Paranoid search. The difference is that at the second ply, not only the moves of the first opponent are considered, but the moves of all opponents are investigated. Similar to paranoid,  $\alpha\beta$ -pruning is possible.

The major disadvantage of incorporating search in the playout phase of MCTS is the reduction of the number of playouts per second [8]. In order to prevent this reduction from outweighing the benefit of the quality of the playouts, enhancements should be implemented to speed up the search and keep the reduction of the number of playouts to a minimum. The number of searches can be reduced by using  $\epsilon$ -greedy playouts [6]. With a probability of  $\epsilon$ , a move is chosen uniform randomly. Otherwise, the selected search technique is used to select the best move. The amount of  $\alpha\beta$ -pruning in a tree can be increased by using *move ordering*. When using move ordering, a player's moves are sorted using a static move evaluator. Another move ordering technique is applying *killer moves* [1]. In each search, two killer moves are always tried first. These are the two last moves that were best or caused a cutoff, at the current depth. Moreover, if the search is completed, the killer moves for that specific level in the playout are stored, such that they can be used during the next MCTS iterations. Killer moves are only used with search techniques where  $\alpha\beta$ -pruning is possible, i.e., Paranoid and BRS search. The size of the tree can be further reduced by using *k-best pruning*. Only the  $k$  best moves are investigated. This reduces the branching factor of the tree from  $b$  to  $k$ .

To test the performance of Playout Search, we performed several round-robin tournaments where each participating player uses a different playout strategy. These playout strategies include 2-ply max<sup>n</sup>, 2-ply Paranoid and 2-ply BRS. Additionally, we include players with one-ply and the static move evaluator as reference players. The tournaments were run for 3-player and 4-player Chinese Checkers and 3-player and

---

<sup>1</sup>The full version of this paper is published in: *Advances in Computer Games (ACG13)*, LNCS 7168, pp. 72–83, 2012.

4-player Focus. In each game, two different player types participate. If one player wins, a score of 1 is added to the total score of the corresponding player type.

In the first set of experiments, all players were allowed to perform 5000 playouts per move. For 3-player Chinese Checkers, BRS is the best technique. It performs slightly better than  $\max^n$  and Paranoid. BRS wins 53.4% of the games against  $\max^n$  and 50.9% against Paranoid. These three techniques perform significantly better than one-ply and the move evaluator. In the 4-player variant,  $\max^n$ , Paranoid and BRS remain the best techniques, where BRS performs slightly better than the other two. BRS wins 53.8% of the games against Paranoid and 51.9% against  $\max^n$ . For 3-player Focus, the best technique is BRS, winning 54.8% against  $\max^n$  and 55.5% against Paranoid.  $\max^n$  and Paranoid are equally strong. BRS is also the best technique in 4-player Focus, though it is closely followed by  $\max^n$  and Paranoid. BRS wins 51.5% of the games against  $\max^n$  and 51.8% against Paranoid.

In the second set of experiments, we gave each player 5 seconds per move. In 3-player Chinese Checkers, one-ply and Paranoid are the best techniques. Paranoid wins 49.2% of the games against one-ply and 68.5% against the move evaluator. BRS ranks third, and the move evaluator and  $\max^n$  are the weakest techniques. In 4-player Chinese Checkers, one-ply is the best technique, closely followed by Paranoid. Paranoid wins 46.3% of the games against one-ply. Paranoid is still stronger than the move evaluator, winning 64.6% of the games. BRS comes in third place, outperforming  $\max^n$  and the move evaluator. One-ply also performs the best in 3-player Focus. Paranoid plays slightly stronger than the move evaluator, with Paranoid winning 51.9% of the games against the move evaluator and 46.1% against one-ply. The move evaluator and Paranoid perform better than BRS and  $\max^n$ . In 4-player Focus, Paranoid performs better than in the 3-player version and slightly outperforms one-ply. Paranoid wins 51.7% of the games against one-ply and 59.9% against the move evaluator.  $\max^n$  also performs significantly better than in the 3-player version. It is as strong as one-ply and better than the move evaluator.

In the final set of experiments, we gave the players 30 seconds per move. Because these games take quite some time to finish, only the one-ply player and the Paranoid player were matched against each other. In the previous set of experiments, these two techniques turned out to be the strongest. Paranoid appears to perform slightly better when the players receive 30 seconds per move compared to 5 seconds per move. In 3-player Chinese Checkers, Paranoid wins 53.9% of the games, compared to 49.2% with 5 seconds. In 4-player Chinese Checkers, 48.3% of the games are won by Paranoid, compared to 46.3% with 5 seconds. In 3-player Focus, the win rate of Paranoid increases from 46.1% with 5 seconds to 50.7% with 30 seconds and in 4-player Focus from 51.7% to 54.1%.

The results show that Playout Search significantly improves the quality of the playouts in MCTS. This benefit is countered by a reduction of the number of playouts per second. Especially BRS and  $\max^n$  suffer from this effect. Based on the experimental results we may conclude that Playout Search for multi-player games might be beneficial if the players receive sufficient thinking time and Paranoid search is employed. Under these conditions, Playout Search outperforms playouts using light heuristic knowledge in the 4-player variant of Focus and the 3-player variant of Chinese Checkers.

## References

- [1] S.G. Akl and M.M. Newborn. The Principal Continuation and the Killer Heuristic. In *Proceedings of the ACM Annual Conference*, pages 466–473, New York, NY, USA, 1977. ACM.
- [2] R. Coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In H.J. van den Herik, P. Ciancarini, and H.H.L.M. Donkers, editors, *Computers and Games (CG 2006)*, volume 4630 of *LNCS*, pages 72–83, Berlin, Germany, 2007. Springer.
- [3] L. Kocsis and C. Szepesvári. Bandit Based Monte-Carlo Planning. In J. Fürnkranz, T. Scheffer, and M. Spiropoulos, editors, *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 282–293, Berlin, Germany, 2006. Springer.
- [4] C. Luckhardt and K.B. Irani. An Algorithmic Solution of N-Person Games. In *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI)*, volume 1, pages 158–162, 1986.
- [5] M.P.D. Schadd and M.H.M. Winands. Best Reply Search for Multiplayer Games. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(1):57–66, 2011.
- [6] N.R. Sturtevant. An Analysis of UCT in Multi-player Games. In H.J. van den Herik, X. Xu, Z. Ma, and M.H.M. Winands, editors, *Computers and Games (CG 2008)*, volume 5131 of *LNCS*, pages 37–49, Berlin, Germany, 2008. Springer.
- [7] N.R. Sturtevant and R.E. Korf. On Pruning Techniques for Multi-Player Games. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 201–207. AAAI Press / The MIT Press, 2000.
- [8] M.H.M. Winands and Y. Björnsson.  $\alpha\beta$ -based Play-outs in Monte-Carlo Tree Search. In *2011 IEEE Conference on Computational Intelligence and Games (CIG 2011)*, pages 110–117. IEEE Press, 2011.

# Influence-Based Abstraction for Multiagent Systems

Frans A. Oliehoek<sup>a</sup>

Stefan J. Witwicki<sup>b</sup>

Leslie P. Kaelbling<sup>c</sup>

<sup>a</sup> *Maastricht University, P.O. Box 616, 6200 MD Maastricht*

<sup>b</sup> *INESC-ID / IST, UTL, Av. Prof. Dr. Cavaco Silva 2744-016 Porto Salvo, Portugal*

<sup>c</sup> *MIT, 32 Vassar Street Cambridge, MA 02139-4307, USA*

## Abstract

The paper presents a theoretical advance by which certain multiagent systems can be decomposed into local models. We formalize the interface between such local models as the *influence* agents can exert on one another. The resulting influence-based abstraction (IBA) generalizes and provides insight into relationships among previous work on exploiting weakly-coupled agent interaction structures. More importantly, given that these previous approaches showed promising increases in efficiency, the current results serve as the foundation for algorithms that we anticipate will bring similar gains to more general planning contexts.

The full version has appeared in *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence*.

## 1 Introduction

This paper focuses on decision-theoretic planning for multiagent systems under uncertainty, as formalized by extensions of the *partially observable MDP (POMDP)*. The generalization to multiple cooperative agents is called *decentralized POMDP (Dec-POMDP)*, and to self-interested agents, the *partially observable stochastic game (POSG)*. To combat the high computational complexity of these models, over the past decade multiagent planning research has branched out into the development of a plethora of more specialized models (as listed in [4]) aimed at gaining computational traction by leveraging various types of agent interaction structure. We propose to unify many of these separate branches with a single theoretical framework, *influence-based abstraction for factored POSGs (IBA)*, in order to exploit weakly-coupled interaction structure in the most general multiagent setting.

## 2 POSGs, Local-Form Models & Influence-Based Abstraction

The starting point of our investigation is the *partially observable stochastic game (POSG)* [1] with a factored state space, which we dub factored POSG (fPOSG). In a (f)POSG, it is possible to use dynamic programming to compute a best response [3]: given the fixed policy of the other agents  $\pi_{-i}$ , from the perspective of agent  $i$  the problem reduces to a single-agent POMDP whose states are pairs  $\langle s^t, \bar{\theta}_{-i}^t \rangle$  and of which the value is the best-response value  $V_i^*(\pi_{-i})$  for agent  $i$ . Beliefs  $b_i^t(s^t, \bar{\theta}_{-i}^t)$  can be maintained by application of Bayes' rule and the solution of the POMDP will provide a value function mapping these beliefs to values.

While the aforementioned *global* individual beliefs constitute a sufficient statistic for computing best-response value, they have the drawback that the agent needs to reason about the entire global state of the system and even the internal state of the other agents. Shown in Fig. 1 (left), we introduce the concept of local form for a POSG, which we term the Local-Form Model (LFM), as the basis of a local, and potentially more compact, belief that is also a sufficient statistic. Intuitively, an agent may not need to reason about the entire state space; its reward function may depend on a small number of state variables, called *state factors*, and so may its observations. It is often quite natural to specify a subset of state factors that will constitute the *local state* of an agent (distinguished by the dotted lines in Figure 1). This is formalized by local form: an LFM is an fPOSG together with a suitable *local state function*  $S$ , which maps from agents to (possibly overlapping) subsets of state factors  $S(i)$  that are in the agents' local states.

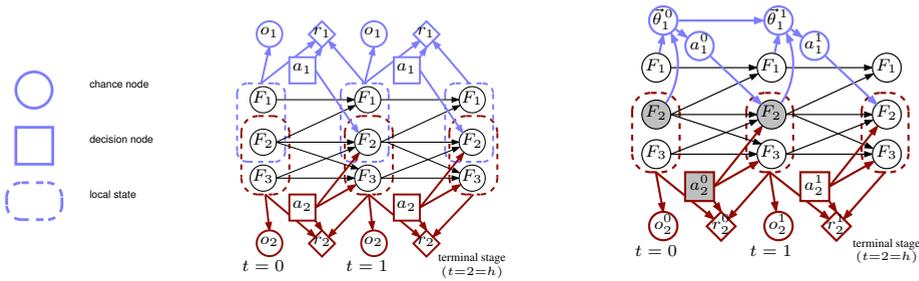


Figure 1: (Left) local form of a factored POSG. (Right) agent 2's best response model.

Using influence-based abstraction, it becomes unnecessary to reason about the entire state space and every detail of other agents' actions; the agent only needs to know how the 'external' portion of the problem (i.e., state factors outside of agent  $i$ 's local state as well as actions and observations of other agents) affects its local state. In Figure 1 (right), we treat an LFM from the perspective of one agent and consider how that agent is affected by the other agents and can compute a best response against that 'incoming influence'. We formalize this *influence*  $I$  as the conditional probability distribution over the *influence sources*, which are the (external) sources of the arrows leading into agent  $i$ 's local state. We have proven [4] that  $I$  need only be conditional on a set  $\vec{D}_i^{t-1}$  of variables that d-separate the influence sources from the local states, actions and observations of agent  $i$ , illustrated in Fig. 1(right) by shading.

IBA is a two-stage process. First the agent computes, for a particular  $\pi_{-i}$ , the incoming influence  $I_{\rightarrow i}(\pi_{-i})$  using (approximate) inference methods [2]. Next, the agent uses this influence, to construct an *influence-augmented local model (IALM)* that replaces the conditional probability tables (CPTs) for all factors that are influence destinations by *induced CPTs* that only depend on the local state and  $\vec{D}_i^{t-1}$ , thereby isolating its local problem from superfluous external variables. Since an IALM is just a special case of POMDP, regular POMDP solution methods can be used to compute  $V_i^*(I_{\rightarrow i}(\pi_{-i}))$ , the value of the IALM. We show that the solution of the IALM for the incoming influence point  $I_{\rightarrow i}(\pi_{-i})$  associated with any  $\pi_{-i}$  achieves the same value as the best response computed against  $\pi_{-i}$  directly:  $\forall \pi_{-i} \quad V_i^*(I_{\rightarrow i}(\pi_{-i})) = V_i^*(\pi_{-i})$ , which means that an IALM can be used to compute a best response.

### 3 Conclusions

This paper has introduced influence-based abstraction for factored POSGs, a technique that allows us to decouple the model into a set of local models defined over subsets of state factors. Performing IBA for an agent consists of two steps: computing the incoming influence point using inference techniques and creating the agent's induced local model. IBA for fPOSGs generalizes the abstractions made in a number of more specific problem contexts [4]. Our formalism is the foundation for future development of influence-space search in general Dec-POMDPs, and of efficient influence-based solution methods for self-interested agents.

### Acknowledgments

Research supported by AFOSR MURI project #FA9550-09-1-0538, by NWO CATCH project #640.005.003, and by the Fundação para a Ciência e a Tecnologia (FCT) and the CMU-Portugal Program under project CMU-PT/SIA/0023/2009.

### References

- [1] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, pages 709–715, 2004.
- [2] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [3] Ranjit Nair, Milind Tambe, Makoto Yokoo, David V. Pynadath, and Stacy Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI*, pages 705–711, 2003.
- [4] Frans A. Oliehoek, Stefan Witwicki, and Leslie P. Kaelbling. Influence-based abstraction for multiagent systems. In *AAAI*, pages 1422–1428, July 2012.

# Enhancements for Monte-Carlo Tree Search in Ms Pac-Man<sup>1</sup>

Tom Pepels

Mark H.M. Winands

*Games and AI Group, Department of Knowledge Engineering, Maastricht University,  
Maastricht, The Netherlands*

Ms Pac-Man is a real-time arcade game based on the popular Pac-Man game. The player controls the main character named Ms Pac-Man (henceforth named *Pac-Man*) through a maze, eating pills and avoiding the ghosts chasing her. The maze contains four so-called power pills that allow the player to eat the ghosts to obtain a higher score. When all pills in a maze are eaten, the game progresses to the next level. Ms Pac-Man inherited its game-mechanics from the original Pac-Man. Moreover, it introduced four different mazes, and more important, unpredictable ghost behaviour. This last feature makes Ms Pac-Man an interesting subject for AI research. The game rules are straightforward, however complex planning and foresight are required for a player to achieve high scores.

Two competitions are held for autonomous Pac-Man agents. In the first, *Ms Pac-Man Competition (screen-capture version)*, the original version of the game is played using an emulator. Agents interpret a capture of the screen to determine the game's state. Each turn moves are passed to the emulator running the game. The second, *Ms Pac-Man vs Ghost Competition* offers a complete implementation of the game, therefore the screen does not need to be captured by the agents, and the game state is fully available. Furthermore, Pac-Man agents compete with a variety of ghost-team agents also entering the competition.

Although most Pac-Man agents entering the competitions are rule-based, research has been performed on using techniques such as genetic programming, neural networks and search trees. Owing to the successful application of Monte-Carlo Tree Search (MCTS) [1, 4] in other games, interest in developing MCTS agents for Ms Pac-Man has grown. Samothrakis *et al.* [5] developed an MCTS agent using a Max<sup>n</sup> tree with scoring for both Pac-Man and the ghosts. Furthermore, a target location is set as a long-term goal for Pac-Man, MCTS computes the optimal route to the target in order to determine the next move. Other MCTS-based agents were researched for achieving specific goals in Ms Pac-Man, such as ghost avoidance [8] and endgame situations [7] demonstrating the possibilities of MCTS for Pac-Man agents. In 2011 the first MCTS agent won the *Ms Pac-Man screen-capture competition*. Until then rule-based agents led the competitions. The victorious MCTS agent, Nozomu [3], was designed to avoid so-called 'pincer moves', in which every escape path for Pac-Man is blocked. The approach was successful in beating the leading rule-based agent ICE Pambush [6] with a high score of 36,280.

An agent for the *Ms Pac-Man vs Ghost Competition* is developed for this extended abstract, therefore no assumptions are made on the ghost-team's behaviour. To enhance the MCTS framework for Pac-Man five enhancements are introduced: 1) a variable depth tree, 2) playout strategies for the ghost-team and Pac-Man, 3) including long-term goals in scoring, 4) endgame tactics, and 5) a Last-Good-Reply policy [2] for memorizing rewarding moves during playouts.

The discussed enhancements for the MCTS framework have resulted in a Pac-Man agent achieving a high score of 127,945 points versus the LEGACY2THERECKONING ghost team. Regarding the results of previous competitions, an average performance gain of 40,962 points, based on the top scoring Pac-Man agent of the CIG'11, is achieved by our MCTS agent. Additional experiments reveal that the variable depth tree and strategic playout are responsible for most of the increase in performance (see Table 1). Although the endgame tactics and Last-Good-Reply policy did not increase the performance significantly, they may be crucial to competing with more advanced ghost teams. However, it is possible that when the playout strategy is further improved, LGR will have less effect on overall scores. Based on the results we may conclude that the MCTS framework makes strong Pac-Man agents possible. Finally, we would like to remark that MCTS

---

<sup>1</sup>The full version of this paper is accepted for *IEEE Conference on Computational Intelligence and Games (CIG-2012)*.

Table 1: Disabled enhancements with the average, maximum and minimum scores based on 100 games

Ghost Team: LEGACY2 T.R., Pac-Man agent: MCTS PAC-MAN				
Enhancement disabled	Avg. score	Max. score	Min. score	95% conf. int.
Strategic playout	44,758	65,270	11,900	2,310
Var. depth tree	101,836	124,925	43,595	3,326
Last-Good-Reply	105,723	125,885	45,830	2,964
Endgame tactic	108,020	125,440	40,945	2,551
MCTS PAC-MAN	107,561	127,945	40,495	2,791

PAC-MAN entered the *Ms Pac-Man vs Ghost team competition* held for the WCCI 2012 under the nickname MAASTRICHT. The agent finished second out of 63 participants.

## References

- [1] R. Coulom. Efficient selectivity and backup operators in Monte-Carlo tree search. In H. J. van den Herik, P. Ciancarini, and H. H. L. M. Donkers, editors, *Computers and Games (CG 2006)*, volume 4630 of *Lecture Notes in Computer Science (LNCS)*, pages 72–83. Springer-Verlag, Heidelberg, Germany, 2007.
- [2] P.D. Drake. The last-good-reply policy for Monte-Carlo Go. *International Computer Games Association Journal*, 32(4):221–227, 2009.
- [3] N. Ikehata and T. Ito. Monte-Carlo tree search in Ms. Pac-Man. In *IEEE Conference on Computational Intelligence and Games (CIG)*, pages 39–46. IEEE, 2011.
- [4] L. Kocsis and C. Szepesvári. Bandit Based Monte-Carlo Planning. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Artificial Intelligence*, pages 282–293, 2006.
- [5] S. Samothrakis, D. Robles, and S.M. Lucas. Fast approximate max-n Monte-Carlo tree search for Ms Pac-Man. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(2):142–154, 2011.
- [6] R. Thawonmas and T. Ashida. Evolution strategy for optimizing parameters in Ms Pac-Man controller ICE Pambush 3. In *IEEE Conference on Computational Intelligence and Games (CIG)*, pages 235–240, 2010.
- [7] B.K.B. Tong, C.M. Ma, and C.W. Sung. A Monte-Carlo approach for the endgame of Ms. Pac-Man. In *IEEE Conference on Computational Intelligence and Games (CIG)*, pages 9–15. IEEE, 2011.
- [8] B.K.B. Tong and C.W. Sung. A Monte-Carlo approach for ghost avoidance in the Ms. Pac-Man game. In *International IEEE Consumer Electronics Society’s Games Innovations Conference (ICE-GIC)*, pages 1–8. IEEE, 2011.

# Toward machines that behave ethically better than humans do (extended abstract)<sup>1</sup>

Matthijs A. Pontier, Johan F. Hoorn

*VU University Amsterdam, Center for Advanced Media Research Amsterdam (CAMERA@VU), De Boelelaan 1081, 1081HV Amsterdam, The Netherlands  
m.a.pontier@vu.nl, j.f.hoorn@vu.nl  
<http://camera-vu.nl/matthijs/>*

In view of increasing intelligence and decreasing costs of artificial agents and robots, organizations increasingly use such systems for more complex tasks. As the intelligence of machines increases, the amount of human supervision decreases and machines increasingly operate autonomously. These developments request that we should be able to rely on a certain level of ethical behavior from machines. As Rosalind Picard [6] nicely puts it: “the greater the freedom of a machine, the more it will need moral standards”. Especially when machines interact with humans, which they increasingly do, we need to ensure that these machines do not harm us or threaten our autonomy.

There have been various approaches in giving machines moral standards, using various methods. One of them, called casuistry, looks at previous cases in which there is agreement about the correct response (e.g., [4]). Using the similarities with these previous cases and the correct responses to them, the machine attempts to determine the correct response to a new ethical dilemma.

However, reclassification of cases remains problematic in these approaches due to a lack of reflection and explicit representation. Therefore, Guarini [4] concludes that casuistry alone is not sufficient. Anderson and Anderson [2] agree with this conclusion, and address the need for top-down processes. The two most dominant top-down mechanisms are (1) utilitarianism and (2) ethics about duties. Utilitarians claim that ultimately morality is about maximizing the total amount of ‘utility’ (a measure of happiness or well being) in the world. The competing ‘big picture’ view of moral principles is that ethics is about duties and, on the flip side of duties, the rights of individuals. The two competitors described above may not differ as much as it seems. Ethics about duties can be seen as a useful model to maximize the total amount of utility.

The current paper can be seen as a first attempt in combining a bottom-up and top-down approach. It combines a bottom-up structure with top-down knowledge in the form of moral duties. It balances between these duties and computes a level of morality, which could be seen as an estimation of the influence on the total amount of utility in the world.

We developed a moral reasoner that combines a bottom-up structure with top-down knowledge in the form of moral duties. The reasoner estimates the influence of an action on the total amount of utility in the world by the believed contribution of the action to the following three duties: Autonomy, Non-maleficence and Beneficence. Following these three duties is represented as having three moral goals. The moral reasoner is capable of balancing between conflicting moral goals. In simulation experiments, the reasoner reached the same conclusions as expert ethicists [3].

Even between doctors, there is no consensus about the interpretation of values and their ranking and meaning. In the work of Van Wynsberghe [8] this differed depending on: the type of care (i.e., social vs. physical care), the task (e.g., bathing vs. lifting vs. socializing), the care-giver and their style, as well as the care-receiver and their specific needs.

---

<sup>1</sup> The full version of this paper appeared in: Proceedings of the 34th International Annual Conference of the Cognitive Science Society, CogSci'12 (2012)

According to Anderson and Anderson [2], fully autonomous decisions should never be questioned. However, it can be questioned whether a patient can ever be fully autonomous. Moreover, it seems that medical ethics are contradictory with the law. In the case of euthanasia, the patient makes a fully autonomous decision that will lead to his death. However, in many countries, committing active euthanasia is illegal. In countries where euthanasia is permitted, it is usually only allowed when the patient is in hopeless suffering. By the definition of Anderson and Anderson, being in hopeless suffering would mean the patient is not free of internal constraints (i.e., pain and suffering) and therefore not capable of making fully autonomous decisions. On the other hand, in the case of hopeless suffering, it could be questioned whether one could speak of maleficence when euthanasia is allowed.

However, we would not like to argue against strict ethical codes in professional fields such as health care. It is important to act based on a consensus to prevent conflicts and unnecessary harm. Just as doctors restrict their ‘natural’ behavior by maintaining a strict ethical code, we can also let a robot restrict its behavior by acting through the same strict ethical code.

Moreover, we may well want to aim for machines that behave ethically better than human beings. Human behavior is typically far from being morally ideal, and a machine should probably have higher ethical standards [1]. By matching the ethical decision-making of expert ethicists, the presented moral reasoner serves as a nice starting point in doing so.

Wallach, Franklin and Allen [9] argue that even agents who adhere to a deontological ethic or are utilitarians may require emotional intelligence as well as other ‘‘supra-rational’’ faculties, such as a sense of self and a theory of mind (ToM). Additionally, Tronto [7] argues that care is only thought of as good care when it is personalized. Therefore, in future research we intend to integrate the moral reasoner with Silicon Coppélia [5], a cognitive model of emotional intelligence and affective decision making. Silicon Coppélia contains a feedback loop, by which it can learn about the preferences of an individual patient, and personalize its behavior.

### Acknowledgements

This study is part of the SELEMCA project within CRISP (grant number: NWO 646.000.003).

### References

- [1] Allen, C. Varner, G. & Zinser, J. (2000). Prolegomena to Any Future Artificial Moral Agent. *Journal of Experimental and Theoretical Artificial Intelligence*, 12, 251–61
- [2] Anderson, M., & Anderson, S. (2007). Machine ethics: Creating an ethical intelligent agent, *AI Magazine*, 28(4), 15-26.
- [3] Buchanan, A.E. & Brock, D.W. (1989). *Deciding for Others: The Ethics of Surrogate Decision Making*, Cambridge University Press.
- [4] Guarini, M. (2006). Particularism and the Classification and Reclassification of Moral Cases. *IEEE Intelligent Systems*, 21(4), 22–28.
- [5] Hoorn, J.F., Pontier, M.A., & Siddiqui, G.F. (2012). Coppélius' Concoction: Similarity and Complementarity Among Three Affect-related Agent Models. *Cognitive Systems Research Journal*, 33-49
- [6] Picard R (1997) *Affective computing*. MIT Press, Cambridge, MA
- [7] Tronto, J. (1993). *Moral Boundaries: a political argument for an ethic of care*. Routledge, New York.
- [8] Van Wynsberghe, A. (2012). Designing Robots for Care; Care Centered Value-Sensitive Design. *Journal of Science and Engineering Ethics*, in press
- [9] Wallach, W., Franklin, S. & Allen, C. (2010). A Conceptual and Computational Model of Moral Decision Making in human and Artificial Agents. *Topics in Cognitive Science*, 2, 454–485.

# ***k*-Optimal: A Novel Approximate Inference Algorithm for ProbLog**

Joris Renkens

Guy Van den Broeck

Siegfried Nijssen

Department of Computer Science, Katholieke Universiteit Leuven

## **Abstract**

ProbLog is a probabilistic extension of Prolog. Key features of ProbLog are that (1) for each clause a probability can be specified that the clause belongs to the program; (2) the success probability of a query is defined by the probability that it succeeds in a sample of the program. Given the high complexity of exactly calculating the success probability of a query under ProbLog's semantics, in many applications approximate inference is necessary. Current approximate inference algorithms for ProbLog however require either dealing with large numbers of proofs or do not guarantee a low approximation error. This paper gives the main insights of our previous work [5] in which we introduce a new approximate inference algorithm which addresses these shortcomings. Given a user-specified parameter  $k$ , this algorithm approximates the success probability of a query based on at most  $k$  proofs and ensures that the calculated probability  $p$  is  $(1 - 1/e)p^* \leq p \leq p^*$ , where  $p^*$  is the highest probability that can be calculated based on any set of  $k$  proofs. Our experiments show the utility of the proposed algorithm.

## **1 Introduction**

The probabilistic logic ProbLog [2] has been used to solve learning problems in probabilistic networks as well as other types of probabilistic data. The key feature of ProbLog is its distribution semantics. Each fact in a ProbLog program can be annotated with the probability that this fact is true in a random sample from the program. The success probability of a query is equal to the probability that the query succeeds in a sample from the program, where facts are sampled independently from each other.

The main problem in calculating the success probability of a query in ProbLog is the high computational complexity of exact inference. As multiple proofs for a query can be simultaneously true, we cannot calculate the success probability of a query based on the probabilities of the independent proofs; we need to deal with a *disjoint sum problem* [2]. This problem becomes worse as the number of proofs grows.

To deal with this computational issue, several approaches have been proposed in the past. One such approach uses Binary Decision Diagrams (BDDs) [2, 1]. BDDs can be seen as a representation of proofs from which the required success probability can be calculated in polynomial time. Building a BDD for all proofs of a query can however be intractable. Because of this, compiling large numbers of proofs into a BDD should be avoided.

The *k*-best strategy avoids this by selecting only  $k$  proofs for constructing the BDD [3]. However, the selection criterion of *k*-best provides few guarantees with respect to the quality of the calculated success probability. In our work [5] we propose *k*-optimal, which has a different selection criterion that gives guarantees for the quality of the calculated success probability.

## **2 *k*-Best and *k*-Optimal**

*k*-Best selects  $k$  proofs based on their individual probability. Unfortunately, a success probability calculated based on these proofs is not necessarily a good approximation of the true probability. Bad approximations occur when there is a high degree of overlap between the selected proofs.

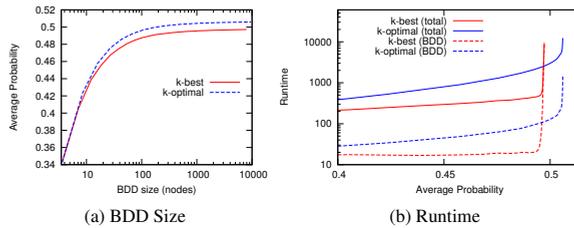


Figure 1: Average probability for increasing BDD size and average runtime in function of the average achieved probability with varying  $k$  values. Runtimes for building the BDDs and total runtimes are shown.

In this paper we propose a new algorithm,  $k$ -optimal, for finding a set of at most  $k$  proofs.  $k$ -Optimal is a greedy algorithm in which in each iteration the proof  $pr$  is added that maximizes  $P(A \cup \{pr\})$  (Algorithm 1), where  $A$  is the set of proofs selected to calculate the success probability from. We use a new and efficient algorithm for calculating  $P(A \cup \{pr\})$  without computing a BDD for each  $A \cup \{pr\}$ .

The key distinguishing feature with respect to  $k$ -best is that if  $p^*$  is the best probability that can be calculated based on  $k$  proofs, our algorithm will not calculate a probability that is worse than  $(1 - 1/e)p^*$ . At the same time, our algorithm ensures that the resulting set of proofs is diverse, i.e., proofs are less likely to use similar facts.

### 3 Results

We evaluated our method on the probabilistic network constructed in [4], which contains 20715 edges and 5313 nodes. This biological network represents the regulatory system of a yeast cell; biologists are interested in pathways that explain the effect of one protein on the expression of another. For this purpose, the connection probability for many pairs of nodes needs to be calculated.

Figure 1a compares  $k$ -best with  $k$ -optimal in terms of BDD size. Each point represents the average probability over multiple queries for a single  $k$  value, with  $k$  varying between 1 and 23. The results clearly show that  $k$ -optimal achieves better approximations of the probabilities with the same BDD size.

Figure 1b shows the average computation time and BDD construction time in function of the average probability for varying  $k$  values. When we are using low  $k$  values, the time that is needed to compute the BDDs is not dominant and  $k$ -best achieves better results due to lower search time. However, with high  $k$  values, the BDD construction time grows exponentially and the smaller BDDs result in a major speed up.

**Acknowledgements** Guy Van den Broeck and Siegfried Nijssen are supported by the Research Foundation-Flanders (FWO-Vlaanderen). This work is also supported by CoE NATAR IOK-C1895-PF/10/010.

### References

- [1] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 35:677–691, 1986.
- [2] L. De Raedt, A. Kimmig, and H. Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *IJCAI*, pages 2462–2467, 2007.
- [3] A. Kimmig, V. Santos Costa, R. Rocha, B. Demoen, and L. De Raedt. On the efficient execution of problog programs. In *ICLP*, pages 175–189, 2008.
- [4] O. Ourfali, T. Shlomi, T. Ideker, E. Ruppim, and R. Sharan. Spine: a framework for signaling-regulatory pathway inference from cause-effect experiments. *Bioinformatics*, 23:359–366, 2007.
- [5] J. Renkens, G. Van den Broeck, and S. Nijssen.  $k$ -Optimal: A novel approximate inference algorithm for ProbLog. In *ILP 2011*, pages 1–6, July 2011.

---

#### Algorithm 1 *greedy\_solve(V)*

---

```

A ← ∅
for i = 1..k do
    A ← A ∪ arg maxpr ∈ V P(A ∪ {pr})
return A

```

---

# Tree-Based Solution Methods for Multiagent POMDPs with Delayed Communication

Matthijs T.J. Spaan <sup>a</sup>

Frans A. Oliehoek <sup>b</sup>

<sup>a</sup> *Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands*

<sup>b</sup> *Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands*

## Abstract

Multiagent Partially Observable Markov Decision Processes (MPOMDPs) provide a powerful framework for optimal decision making under the assumption of instantaneous communication. We focus on a delayed communication setting (MPOMDP-DC), in which broadcasted information is delayed by at most one time step. In this paper, we show that computation of the MPOMDP-DC backup can be structured as a tree and we introduce two novel *tree-based pruning* techniques that exploit this structure in an effective way.

The full version of this paper has appeared in *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence (AAAI-12)*.

## 1 Introduction

Planning under uncertainty in multiagent systems can be neatly formalized as a *decentralized partially observable Markov decision process (Dec-POMDP)*, but solving a Dec-POMDP is a complex (NEXP-complete) task. Communication can mitigate some of these complexities; by allowing agents to share their individual observations the problem reduces to a so-called *multiagent POMDP (MPOMDP)*, a special instance of the standard POMDP [2] which is in PSPACE. However, this model requires the agents to perform full synchronization of their knowledge before selecting a next action, which is inappropriate in domains in which agents may need to act fast in response to their individual observations.

A prime example is decentralized protection control in electricity distribution networks by so-called Intelligent Electronic Devices (IED). As power grids move towards integrating more distributed generation capability (e.g., provided by solar panels or fuel cells), more intricate protection schemes have to be developed as power flow is no longer unidirectional. In response, modern IEDs not only decide based on locally available sensor readings, but can receive information from other IEDs through a communication network with deterministic delays. When extreme faults such as circuit or cable failures occur, however, no time can be wasted waiting for information from other IEDs to arrive.

## 2 Tree-based pruning methods for MPOMDP-DC models

In this paper we focus on a class of problems where agents share their individual observations with a one step delay. That is, agents act using a *one step delayed sharing pattern*, resulting in an *MPOMDP with delayed communication (MPOMDP-DC)*. Solutions for such settings are also useful under longer delays [4]. Moreover, this class is particularly interesting, because it avoids the delay in action selection due to synchronization, while it is very similar to the standard POMDP. However, even though dynamic programming algorithms date back to the eighties, computational difficulties have limited the model's applicability.

The MPOMDP-DC value function is piecewise-linear and convex over the joint belief space, which is a property exploited by many regular POMDP solvers. However, *incremental pruning (IP)* [1], that performs a key operation, the so-called *cross-sum*, more efficiently, is not directly able to achieve the same improvements under delayed communication. A naive application of this technique (NAIVE IP) needs to loop over a number of decentralized control laws that is exponential both in the number of agents and in the number of observations. We target this additional complexity by proposing two novel methods that operate over a tree structure [3], illustrated in Figure 1a. These methods prune exactly the same vectors as NAIVE IP, but they iterate over the set of candidate vectors in a different way: NAIVE IP loops over all decentralized control laws  $\beta$ , while TBP methods exploit the similar parts in different  $\beta$ . The first method, called TBP-M for tree-based pruning with memoization, avoids *duplicate* work by

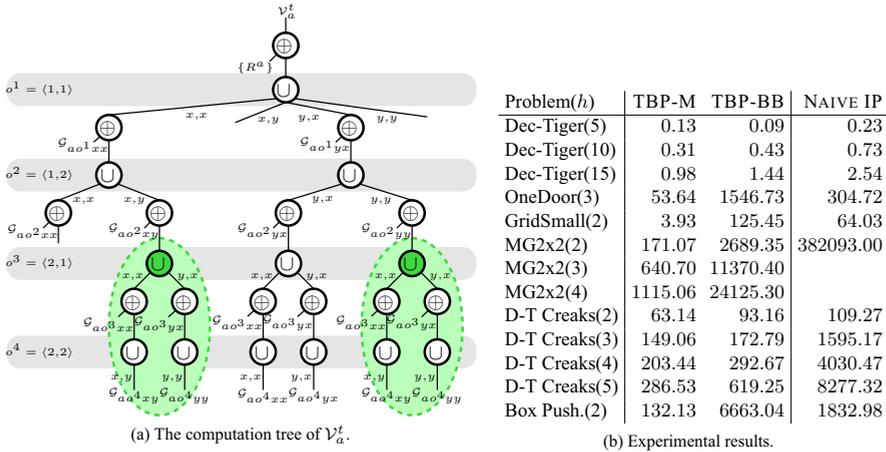


Figure 1: (a) Example of the identified tree structure. (b) Timing results (in s).

caching the result of computations at internal nodes and thus accelerates computation at the expense of memory. The second algorithm, branch and bound (TBP-BB), tries to avoid *unnecessary* computation by making use of upper and lower bounds to prune parts of the tree, providing a different space/time tradeoff.

The empirical evaluation of the proposed methods on a number of test problems shows a clear improvement over NAIVE IP, as presented in Table 1b. TBP-M provides speedups of up to 3 orders of magnitude. TBP-BB does not consistently outperform the baseline, but is still able to provide large speedups on a number of test problems, while using little memory.

### 3 Conclusions

In our work we considered multiagent planning under uncertainty formalized as a multiagent POMDP with delayed communication (MPOMDP-DC). A key feature of this model is that it allows a fast response to certain local observations, relevant in time-critical applications such as intelligent power grid control. We proposed two novel methods for computing optimal control policies in this model, and the experimental results show that we have successfully mitigated the additional complexity that the MPOMDP-DC backup exhibits over the MPOMDP one, allowing for the solution of larger problems.

### Acknowledgments

M.S. is funded by the FP7 Marie Curie Actions Individual Fellowship #275217 (FP7-PEOPLE-2010-IEF). Research supported by AFOSR MURI project #FA9550-09-1-0538 and NWO CATCH project #640.005.003.

### References

- [1] Anthony Cassandra, Michael L. Littman, and Nevin L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *UAI*, pages 54–61. Morgan Kaufmann, 1997.
- [2] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.
- [3] Frans A. Oliehoek and Matthijs T. J. Spaan. Tree-based pruning for multiagent POMDPs with delayed communication. In *AAAI*, 2012.
- [4] Matthijs T. J. Spaan, Frans A. Oliehoek, and Nikos Vlassis. Multiagent planning under uncertainty with stochastic communication delays. In *ICAPS*, pages 338–345, 2008.

# Lifted Variable Elimination with Arbitrary Constraints<sup>1</sup>

Nima Taghipour      Daan Fierens      Jesse Davis      Hendrik Blockeel

*Department of Computer Science, KU Leuven  
Celestijnenlaan 200A, 3001 Heverlee, Belgium*

## Abstract

Lifted inference methods exploit regularities in the structure of probabilistic models: they perform inference once for an entire group of interchangeable objects, instead of for each object in the group. Existing lifted inference methods use a specific constraint language for defining the groups. In this work we generalize lifted variable elimination to work with arbitrary constraints. We empirically demonstrate that this improves inference efficiency by orders of magnitude, allowing exact inference on problems for which until now only approximate inference was feasible.

## 1 Lifted Probabilistic Inference

Probabilistic logical models (PLMs) [1, 3] combine elements of first-order logic with probabilistic graphical models to compactly represent complex, uncertain, structured domains. Formulas in first-order logic can express that some property holds for an entire set of objects, e.g., a formula can state that all humans are mortal:  $\forall X : Human(X) \rightarrow Mortal(X)$ . PLMs can, in a similar way, express probabilistic knowledge about an entire set of objects, e.g., they could state that for each human, there is a prior probability of 20% that he or she smokes:  $\forall X : Human(X) \rightarrow P(Smokes(X)) = 0.2$ .

PLMs can involve a large number of objects (e.g., the population of people in a country), which makes efficient inference a major challenge. First-order logic can reason on the level of logical variables: one can derive  $Mortal(X)$  from  $Human(X)$  without knowing what  $X$  is. Many approaches for reasoning in PLMs, however, transform their knowledge into a propositional graphical model before performing inference. By doing so, they lose the capacity to reason on the level of logical variables: standard inference methods for graphical models can reason only on the “ground” level, repeating the same inference steps for each different instance  $x$  of  $X$ , instead of once for all  $x$ .

Addressing this problem, Poole [7] introduced the concept of lifted inference for graphical models. The basic principle is to group together interchangeable objects, and perform the inference operations once for the entire group instead of once for each individual object in the group. Many different algorithms for lifted inference have been proposed (e.g., [7, 2, 6, 4, 8]). A key factor in the efficiency of these algorithms is the granularity of the grouping: coarser groupings allow more “lifted” operations, leading to faster run times. In this work we aim to improve inference efficiency by performing lifted inference with a coarser granularity.

## 2 The Importance of Constraints

In lifted inference, a group of interchangeable objects is typically defined by means of a constraint that an object must fulfill in order to belong to that group. As such, the type of constraints that are allowed, and the way in which they are handled, directly influence the granularity of the grouping, and hence, the efficiency of the subsequent lifted inference [5].

---

<sup>1</sup>This work was published at the 15th International Conference on Artificial Intelligence and Statistics (AISTATS), 2012, La Palma, Canary Islands.

Many existing approaches for exact lifted inference use a specific class of constraints, namely conjunctions of pairwise (in)equalities. This provides the bare minimum required for lifted inference but unnecessarily limits the symmetries the model can capture, i.e. the groups that can be defined. For instance, with pairwise (in)equalities we can represent a group of objects such as  $(X, Y) | X \neq ann$ , but not a group such as  $(X, Y) \in \{(ann, bob), (bob, carl)\}$ , which prevents the use of lifting for reasoning about this group. We solve this problem by decoupling the lifted inference algorithm from the used constraint language.

### 3 Lifted Inference with Arbitrary Constraints

Our main contribution is the definition of operators for lifted inference that work correctly for any constraint language. For this, we treat constraints simply as sets of tuples (relations), and define the operators using relational algebra operations. The operators thus work on the semantic level (the constraints' extension) rather than on the syntactic level, which makes them language-independent. As a result, they can work with arbitrary constraints, which allows the lifted inference method to capture a broader range of symmetries, leading to more opportunities for lifting.

We also propose a concrete mechanism for representing and manipulating arbitrary constraints. Viewing constraints as sets of tuples does not imply that they must be represented extensionally (which would be inefficient); it only implies that the operators are correct for whatever constraint representation is being used. It is important, however, to handle these constraints efficiently. For this, we represent an arbitrary constraint using a decision tree structure, which we call a *constraint tree*. We show how the constraint handling operators can be implemented with this particular mechanism, i.e., as operators on constraint trees.

The resulting system [9] is more powerful w.r.t. the groups of interchangeable objects it forms, and thus performs lifted inference with a much coarser granularity, than its predecessors. We empirically demonstrate that this can dramatically improve inference efficiency. We evaluate our system on several domains, and show that our approach results in up to three orders of magnitude improvement in runtime. Furthermore, our system can solve several tasks that are intractable for the existing systems, thus allowing exact inference where until now only approximate inference was feasible.

## References

- [1] Luc De Raedt, Paolo Frasconi, Kristian Kersting, and Stephen Muggleton, editors. *Probabilistic inductive logic programming: theory and applications*. Springer-Verlag, Berlin, Heidelberg, 2008.
- [2] Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. Lifted first-order probabilistic inference. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1319–1325, 2005.
- [3] Lise Getoor and Ben Taskar, editors. *An Introduction to Statistical Relational Learning*. MIT Press, 2007.
- [4] Kristian Kersting, Babak Ahmadi, and Sriraam Natarajan. Counting belief propagation. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI-09)*, 2009.
- [5] Jacek Kisynski and David Poole. Constraint processing in lifted probabilistic inference. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI-09)*, 2009.
- [6] Brian Milch, Luke S. Zettlemoyer, Kristian Kersting, Michael Haimes, and Leslie Pack Kaelbling. Lifted probabilistic inference with counting formulas. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, pages 1062–1608, 2008.
- [7] David Poole. First-order probabilistic inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 985–991, 2003.
- [8] Parag Singla and Pedro Domingos. Lifted first-order belief propagation. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, pages 1094–1099, 2008.
- [9] Nima Taghipour, Daan Fierens, Jesse Davis, and Hendrik Blockeel. Lifted variable elimination with arbitrary constraints. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS-12)*. JMLR Workshop and Conference Proceedings, volume 22, 2012.

# N-Grams and the Last-Good-Reply Policy<sup>1, 2</sup> applied in General Game Playing

Mandy J.W. Tak<sup>a</sup>

Mark H.M. Winands<sup>a</sup>

Yngvi Björnsson<sup>b</sup>

<sup>a</sup> *Games and AI Group, Department of Knowledge Engineering, Maastricht University, Maastricht, The Netherlands*

<sup>b</sup> *School of Computer Science, Reykjavík University, Reykjavík, Iceland*

## Introduction

The aim of General Game Playing (GGP) is to create programs capable of playing a wide range of different games at an expert level, given only the rules of the game. The most successful GGP programs currently employ simulation-based Monte-Carlo Tree Search (MCTS) [3, 5]. The performance of MCTS depends heavily on the simulation strategy used in the play-outs [4].

In this abstract we introduce improved simulation strategies that we test in the GGP agent CADIAPLAYER [2], which won the International GGP competition in both 2007 and 2008. There are two aspects to the improvements: first, we show that a simple  $\epsilon$ -greedy [7] exploration strategy works better than the softmax-based Gibbs measure currently used in CADIAPLAYER and, secondly, we introduce a general framework based on N-Grams for learning promising move sequences. We additionally perform experiments with the *Last-Good-Reply Policy*, which has already been shown to be successful in Go and Havannah [1, 6].

## Simulation Strategies

The *Move-Average Sampling Technique* (MAST) [2] was the main simulation strategy used in CADIAPLAYER when winning the International GGP competition in 2008. For each move  $a$ ,  $Q_h(a)$  stores the average of the returned rewards of the play-outs in which move  $a$  occurred. In the original version of MAST, the  $Q_h(a)$  values are used together with the Gibbs measure, which means that moves with a higher  $Q_h(a)$  have a higher probability of being selected. A disadvantage of the Gibbs measure is that the probability of selecting the move with the highest  $Q_h(a)$  value is not fixed and unknown. Consequently, it is not assured that moves with the highest  $Q_h(a)$  value will be even chosen at all. Therefore, for comparison, we have implemented a different exploration technique also based on the  $Q_h(a)$  values, i.e.  $\epsilon$ -greedy [7]. In the experiments, we refer to MAST when Gibbs measure is used and MASTG when  $\epsilon$ -greedy is employed.

MAST generalizes the merits of moves, without considering the context in which the moves are made. Therefore, we introduce the *N-Gram Selection Technique* (NST) which increases the context by keeping track of move sequences instead of single moves. The N-Grams in NST consist of consecutive move sequences  $s_i$  of length  $i \in \{1, 2, 3\}$ . Similar to MAST the average of the returned rewards of the play-outs is accumulated in  $R(s_i)$ . In the play-out, for each legal move  $a$ , a score is calculated by taking the average of the  $R(s_i)$  values of the sequences  $s_i$  that would occur when move  $a$  is played. These scores are used with an  $\epsilon$ -greedy strategy in order to determine the move to select.

The *Last-Good-Reply Policy* (LGRP) [1] is a simulation strategy that has been already applied in Go and Havannah [1, 6]. The aim is to investigate whether this simulation strategy works in a broader range of games. LGRP keeps track of the successful replies that occurred in the play-outs. During a play-out, moves

<sup>1</sup>The full version of this article is accepted for publication in: *IEEE Transactions on Computational Intelligence and AI in Games* Vol.4, No.2, pp. 73-83 (2012)

<sup>2</sup>This work is funded by the Netherlands Organisation for Scientific Research (NWO) in the framework of the project GoGeneral, grant number 612.001.121.

are selected that are stored as a successful reply to the immediately preceding moves. If such a legal move is unavailable, then a default fall-back strategy decides which move to play. In the experiments, this strategy is either MASTG or NST. They are referred to as: LGR-MASTG and LGR-NST.

## Results

A selection of the experimental results is shown in Table 1. Four experiments were performed with 13 games and two additional experiments were carried out with 5 games. The *startclock* is 60 s, which is the time between receiving the rules and the first move. The *playclock* is 30 s, which is the time between each move.

The different simulation strategies introduced above were benchmarked against a random play-out strategy, named UCT. The results show that overall NST has the best performance, indicating that the increased context indeed is beneficial. LGR-MASTG and LGR-NST are also able to improve upon MAST in several games, which shows that LGRP can be applied in other games besides Go and Havannah.

To compare  $\epsilon$ -greedy [7] with Gibbs measure, MASTG was matched against MAST. The 6th column indicates that an  $\epsilon$ -greedy strategy performs better than Gibbs measure. A possible explanation is that with Gibbs measure it is not assured that moves with the highest  $Q_h(a)$  scores will be chosen at all.

The aim of the final experiment is to show how much benefit the two improvements: (1)  $\epsilon$ -greedy and (2) NST give over the original MAST simulation strategy. Therefore, NST was matched against MAST. The 7th column reveals that on a set of 5 games, NST outperforms MAST with a win rate of approximately 70%.

We conclude that NST together with  $\epsilon$ -greedy forms a robust strategy able to improve upon the simulation strategy MAST used by CADIAPLAYER. Furthermore, these enhancements are also shown to be effective in multiplayer and simultaneous-move games.

Game	MAST vs UCT	NST vs UCT	LGR-MASTG vs UCT	LGR-NST vs UCT	MASTG vs MAST	NST vs MAST
Connect5	66.5 ( $\pm 5.34$ )	85.8 ( $\pm 3.95$ )	84.0 ( $\pm 4.15$ )	91.3 ( $\pm 3.18$ )	57.9 ( $\pm 5.50$ )	77.2 ( $\pm 4.75$ )
Checkers	56.7 ( $\pm 5.61$ )	71.0 ( $\pm 5.11$ )	76.9 ( $\pm 4.69$ )	69.6 ( $\pm 5.18$ )	71.1 ( $\pm 5.06$ )	71.5 ( $\pm 5.07$ )
Breakthrough	86.0 ( $\pm 3.93$ )	96.4 ( $\pm 2.10$ )	88.0 ( $\pm 3.68$ )	97.3 ( $\pm 1.82$ )	43.0 ( $\pm 5.46$ )	71.3 ( $\pm 5.12$ )
Othello	70.1 ( $\pm 5.15$ )	82.8 ( $\pm 4.27$ )	70.0 ( $\pm 5.19$ )	73.6 ( $\pm 4.93$ )	64.6 ( $\pm 5.28$ )	73.0 ( $\pm 5.02$ )
Skirmish	45.0 ( $\pm 5.63$ )	71.5 ( $\pm 5.11$ )	56.2 ( $\pm 5.61$ )	69.2 ( $\pm 5.22$ )	61.8 ( $\pm 5.41$ )	74.7 ( $\pm 4.44$ )
Clobber	52.0 ( $\pm 5.65$ )	54.3 ( $\pm 5.64$ )	55.6 ( $\pm 5.21$ )	48.7 ( $\pm 5.64$ )		
Sheep and Wolf	51.3 ( $\pm 5.66$ )	65.3 ( $\pm 5.39$ )	55.3 ( $\pm 5.63$ )	71.0 ( $\pm 5.13$ )		
TTCC4	60.6 ( $\pm 5.05$ )	66.0 ( $\pm 4.89$ )	44.7 ( $\pm 5.14$ )	45.7 ( $\pm 5.15$ )		
Chinese Checkers	65.1 ( $\pm 4.85$ )	73.0 ( $\pm 4.22$ )	59.7 ( $\pm 4.98$ )	59.4 ( $\pm 4.99$ )		
Chinook	79.3 ( $\pm 4.56$ )	90.0 ( $\pm 3.39$ )	84.2 ( $\pm 4.13$ )	88.7 ( $\pm 3.59$ )		
Runners	54.2 ( $\pm 5.64$ )	31.3 ( $\pm 5.25$ )	37.5 ( $\pm 5.48$ )	28.7 ( $\pm 5.12$ )		
Fighter	49.3 ( $\pm 5.66$ )	51.9 ( $\pm 5.46$ )	51.5 ( $\pm 5.66$ )	52.8 ( $\pm 5.48$ )		
Pawn Whopping	74.3 ( $\pm 4.91$ )	74.0 ( $\pm 4.93$ )	71.4 ( $\pm 5.08$ )	73.2 ( $\pm 4.93$ )		

Table 1: Win % of the first player including a 95% confidence bound

## References

- [1] H. Baier and P.D. Drake. The Power of Forgetting: Improving the Last-Good-Reply Policy in Monte Carlo Go. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(4):303–309, December 2010.
- [2] Y. Björnsson and H. Finnsson. CadiaPlayer: A Simulation-Based General Game Player. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(1):4–15, March 2009.
- [3] R. Coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In H. J. van den Herik, P. Ciancarini, and H. H. L. M. Donkers, editors, *CG 2006*, volume 4630 of *LNCS*, pages 72–83, Berlin-Heidelberg, Germany, 2007. Springer-Verlag.
- [4] S. Gelly and D. Silver. Combining Online and Offline Knowledge in UCT. In Z. Ghahramani, editor, *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 273–280, New York, New York, 2007. ACM.
- [5] L. Kocsis and C. Szepesvári. Bandit Based Monte-Carlo Planning. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *Proceedings of the EMCL 2006*, volume 4212 of *LNCS*, pages 282–293, Berlin-Heidelberg, Germany, 2006. Springer-Verlag.
- [6] J. A. Stankiewicz, M. H. M. Winands, and J. W. H. M. Uiterwijk. Monte-Carlo Tree Search Enhancements for Havannah. In H. J. van den Herik and A. Plaat, editors, *Advances in Computer Games (ACG 13)*, volume 7168 of *LNCS*, pages 60–71, Berlin-Heidelberg, Germany, 2012. Springer-Verlag. In Press.
- [7] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts, March 1998.

# WebPIE: A Web-scale Parallel Inference Engine using MapReduce

Jacopo Urbani<sup>a</sup>, Spyros Kotoulas<sup>a</sup>, Jason Maassen<sup>a</sup>, Frank Van Harmelen<sup>a</sup>, Henri Bal<sup>a</sup>

<sup>a</sup> *Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands*  
*Email: {jacopo,kot,jason,frankh,bal}@cs.vu.nl*

## Abstract

The large amount of Semantic Web data and its fast growth pose a significant computational challenge in performing efficient and scalable reasoning. On a large scale, the resources of single machines are no longer sufficient and we are required to distribute the process to improve performance.

The article that we attach to our submission [2] tackles this problem proposing a methodology to perform inference materializing every possible consequence using the MapReduce programming model. We introduce a number of optimizations to address the issues that a naive implementation would raise and to improve the overall performance. We have implemented the presented techniques in a prototype called WebPIE and the evaluation shows that our approach is able to perform complex inference based on the OWL language over a very large input of about 100 billion triples. To the best of our knowledge, it is the only approach that demonstrates complex inference over an input of a hundred billion of triples.

## 1 Introduction

In the Semantic Web, data is normally represented using the RDF data model [1] in the form of *subject-predicate-object* statements which are released and made available on several websites. An example of RDF statement (also called *triple* in the RDF terminology) is

```
<http://www.vu.nl> <rdf:type> <http://dbpedia.org/University> .
```

and it states that the concept represented by the URI “http://www.vu.nl” is of type “http://.../University”.

There are several advantages of representing the information using this data model. One of them is that applications are able to *reason* over these statements in order to infer new knowledge. For example, suppose that on the Web there is published another statement that says that every university is an educational institution. In this case, applications can deduce that the VU is also of type “educational institution” since the VU is an university and every university is an educational institution.

Scalable reasoning is a crucial problem in the Semantic Web. At the beginning of 2009, the Semantic Web was estimated to contain 4.4 billion statements. One year later, the size of the Web had tripled to 13 billion statements and the current trend indicates that this growth rate has not changed. With such growth, reasoning on a Web scale becomes increasingly challenging, due to the large volume of data involved and to the complexity of the task.

Most current reasoners are designed with a centralized architecture where the execution is carried out by a single machine. When the input size is on the order of billions of statements, the machines hardware becomes the bottleneck. This is a limiting factor for performance and scalability. A distributed approach to reasoning is potentially more scalable because its performance can be improved by adding more computational nodes. However, distributed reasoning is significantly more challenging because it requires developing protocols and algorithms to efficiently share both data and computation.

In this paper, we choose to follow a distributed approach to perform rule-based forward reasoning based on the MapReduce programming model. The choice of MapReduce as programming model is motivated by the fact that MapReduce is designed to limit data exchange and alleviate load balancing problems by dynamically scheduling jobs on the available nodes.

## 2 Our approach

Simply encoding the inference rules using MapReduce is not enough in terms of performance, and research is necessary to come up with efficient distributed algorithms. In order to improve the performance, we have introduced a number of key optimizations to handle the set of rules introduced in the OWL Horst fragment. These are:

- Load the schema triples in memory and, when possible, execute the join on-the-fly instead of in the reduce phase;
- Perform the joins during the reduce phase and use the map function to group the triples in order to avoid duplicates;
- Execute the RDFS rules in a specific order to minimize the number of MapReduce jobs.
- Limit duplicates when performing joins between instance triples using contextual information;
- Limit the exponential derivation of owl:sameAs triples building a sameAs table;
- Perform redundant joins to avoid load balancing problems.

Also, we have implemented mechanisms to apply rules with different strategies and added support to perform reasoning on incremental data.

Some of the key assumptions behind our algorithms are: (a) the schema must be small enough to fit in main memory; (b) for rules with multiple joins, some of the joins must be performed in-memory, (c) we assume that there is no ontology hijacking that might cause an explosion of the derivation; (d) all the input is available locally in the distributed filesystem.

To evaluate our method, we have implemented a prototype called WebPIE (Web-scale Parallel Inference Engine) using the Hadoop framework. We have deployed WebPIE on a 64-node cluster as well as on the Amazon cloud infrastructure and we have performed experiments using both real-world and synthetic benchmark data. The obtained results show that our approach can scale to a very large size, outperforming all published approaches, both in terms of throughput and input size by at least an order of magnitude. To the best of our knowledge, it is the only approach that demonstrates complex Semantic Web reasoning for an input of a hundred billion triples.

## 3 Conclusions

In the attached paper, we have implemented a reasoning method using the MapReduce programming paradigm and present a number of optimizations that address the challenges in performing inference of a large scale.

Our choice of MapReduce was mainly made for reasons of performance and scalability. Although it is easy to create artificial datasets which would degrade the performance, we did not observe such cases in realistic data. In fact, the above assumptions (a)(d) could also serve as guidelines in the design of ontologies and datasets, to ensure that they can be used effectively.

The presented technique is optimized for the execution of the OWL-Horst rules. Future work lies in reasoning over user-supplied rulesets, where the system would choose the correct implementation for each rule and the most efficient execution order, depending on the input.

We believe that this paper establishes that computing the closure of a very large centrally available dataset is no longer an important bottleneck, and that research efforts should switch to other modes of reasoning. Query-driven backward-chaining inference over distributed datasets might turn out to be more promising than exhaustive forward inference over centralized stores.

## References

- [1] F. Manola, E. Miller. RDF Primer. *W3C Recommendation* (<http://www.w3.org/TR/rdf-primer/>), 2004.
- [2] J. Urbani, S. Kotoulas, J. Maassen, F. V. Harmelen, and H. Bal. Webpie: A web-scale parallel inference engine using MapReduce. *Journal of Web Semantics*, 10(0):59 – 75, 2012.

# A Framework for Qualitative Multi-Criteria Preferences: Extended Abstract\*

Wietske Visser    Reyhan Aydoğan    Koen V. Hindriks    Catholijn M. Jonker

Interactive Intelligence Group, Delft University of Technology, The Netherlands  
{Wietske.Visser, R.Aydogan, K.V.Hindriks, C.M.Jonker}@tudelft.nl

**Introduction** A key challenge in the representation of qualitative, multi-criteria preferences is to find a compact and expressive representation. Various frameworks have been introduced, each of which with its own distinguishing features. In this paper we introduce a new representation framework called qualitative preference systems (QPS), which combines priority, cardinality and conditional preferences. Moreover, the framework incorporates knowledge that serves two purposes: to impose (hard) constraints, but also to define new (abstract) concepts.

QPSs are based on the lexicographic rule studied in [1]. This rule is a fundamental part of the framework presented as it offers a principled tool for combining basic preferences. We believe this ability to combine preferences is essential for any practical approach to representing qualitative preferences. It is needed in particular for constructing multi-criteria preferences. It is not sufficient, however, since more expressivity is needed and useful in practice. Therefore, QPSs in addition provide a tool for representing knowledge, for abstraction, for counting, and provide a layered structure for representing preference orderings. QPSs are able to represent various strategies for defining preference orderings, and are able to handle conditional preferences. Logical Preference Description language (LPD; [3]) can be embedded into the QPS framework and that there is an order preserving embedding of CP-nets [2] in the QPS framework. These embeddings provide a representation that is just as succinct as the LPD expressions and CP-nets.

**Qualitative Preference Systems** The main aim of a QPS is to determine preferences between *outcomes* in a purely qualitative way. An outcome is an assignment of values to a set of relevant variables. Every variable has its own domain of possible values. Constraints on the assignments of values to variables are expressed in a knowledge base. Outcomes are defined as variable assignments that respect the constraints in the knowledge base. The preferences between outcomes are based on multiple *criteria*. Every criterion can be seen as a *reason* for preference, or as a preference from one particular *perspective*. We distinguish between simple criteria that are based on a single variable and compound criteria that combine multiple criteria in order to determine an overall preference. There are two kinds of compound criteria: lexicographic criteria and cardinality criteria.

**Definition 1. (Qualitative preference system)** A qualitative preference system (QPS) is a tuple  $(Var, Dom, K, C)$ . *Var* is a finite set of variables. Every variable  $X \in Var$  has a domain  $Dom(X)$  of possible values. *K* (a knowledge base) is a set of constraints on the assignments of values to the variables in *Var*. A constraint is an equation of the form  $X = Expr$  where  $X \in Var$  is a variable and *Expr* is an algebraic expression that maps to  $Dom(X)$ . An outcome  $\alpha$  is an assignment of a value  $x \in Dom(X)$  to every variable  $X \in Var$ , such that no constraints in *K* are violated.  $\alpha_X$  denotes the value of variable *X* in outcome  $\alpha$ . *C* is a finite rooted tree of criteria, where leaf nodes are simple criteria and other nodes are compound criteria. Child nodes of a compound criterion are called its subcriteria. Weak preference between outcomes by a criterion *c* is denoted by the relation  $\succeq_c$ .  $\succ_c$  denotes the strict subrelation,  $\approx_c$  the indifference subrelation.

**Simple criteria** A simple criterion specifies a preference ordering on the values of a single variable. Its preference between outcomes is based solely on the value of this variable in the considered outcomes.

---

\*This is an abstract of [4]. More information about the ideas in this abstract and references to relevant literature can be found there.

**Definition 2. (Simple criterion)** A *simple criterion*  $c$  is a tuple  $\langle X_c, \succeq_c \rangle$ , where  $X_c \in \text{Var}$  is a variable, and  $\succeq_c$ , a preference relation on the possible values of  $X_c$ , is a preorder on  $\text{Dom}(X_c)$ .  $\succ_c$  is the strict subrelation,  $\equiv_c$  is the indifference subrelation. We call  $c$  a *Boolean simple criterion* if  $X_c$  is Boolean and  $\top \succ_c \perp$ . A simple criterion  $c = \langle X_c, \succeq_c \rangle$  *weakly prefers* an outcome  $\alpha$  over an outcome  $\beta$ , denoted  $\alpha \succeq_c \beta$ , iff  $\alpha_{X_c} \succeq_c \beta_{X_c}$ .

**Observation 1.** Let  $c = \langle X_c, \succeq_c \rangle$  be a simple criterion. Then  $\succeq_c$  is a preorder. If  $\succeq_c$  is total, then so is  $\succeq_c$ .

**Lexicographic criteria** A lexicographic criterion consists of a set of subcriteria and an associated priority order (a strict partial order, which means that no two subcriteria can have the same priority). It weakly prefers outcome  $\alpha$  over outcome  $\beta$  if for every subcriterion, either this subcriterion weakly prefers  $\alpha$  over  $\beta$ , or there is another subcriterion with a higher priority that strictly prefers  $\alpha$  over  $\beta$ . This definition of preference by a lexicographic criterion is equivalent to the priority operator as defined by [1]. It generalizes the familiar rule used for alphabetic ordering of words, such that the priority can be any partial order and the combined preference relations can be any preorder.

**Definition 3. (Lexicographic criterion)** A *lexicographic criterion*  $c$  is a tuple  $\langle C_c, \triangleright_c \rangle$ , where  $C_c$  is a nonempty set of criteria (the *subcriteria* of  $c$ ) and  $\triangleright_c$ , a *priority relation* among subcriteria, is a strict partial order (a transitive and asymmetric relation) on  $C_c$ . A lexicographic criterion  $c = \langle C_c, \triangleright_c \rangle$  *weakly prefers* an outcome  $\alpha$  over an outcome  $\beta$ , denoted  $\alpha \succeq_c \beta$ , iff  $\forall s \in C_c (\alpha \succeq_s \beta \vee \exists s' \in C_c (\alpha \triangleright_{s'} \beta \wedge s' \triangleright_c s))$ .

**Proposition 1.** Let  $c = \langle C_c, \triangleright_c \rangle$  be a lexicographic criterion. If for all subcriteria  $s \in C_c$ ,  $\succeq_s$  is a preorder, then the relation  $\succeq_c$  is also a preorder.

**Cardinality criteria** Like a lexicographic criterion, a cardinality criterion combines multiple criteria into one preference ordering. Unlike a lexicographic criterion, priority between subcriteria is not a strict partial order, but all subcriteria have the same priority. A cardinality criterion weakly prefers an outcome  $\alpha$  over an outcome  $\beta$  if it has at least as many subcriteria that strictly prefer  $\alpha$  over  $\beta$  as criteria that do not weakly prefer  $\alpha$  over  $\beta$ .

**Definition 4. (Cardinality criterion)** A *cardinality criterion*  $c$  is a tuple  $\langle C_c \rangle$  where  $C_c$  is a nonempty set of criteria (the *subcriteria* of  $c$ ). A cardinality criterion  $c = \langle C_c \rangle$  *weakly prefers* an outcome  $\alpha$  over an outcome  $\beta$ , denoted  $\alpha \succeq_c \beta$ , iff  $|\{s \in C_c \mid \alpha \triangleright_s \beta\}| \geq |\{s \in C_c \mid \alpha \not\succeq_s \beta\}|$ .

**Proposition 2.** Let  $c = \langle C_c \rangle$  be a cardinality criterion such that for all  $s \in C_c$ ,  $s$  is a Boolean simple criterion. Then  $\succeq_c$  is a preorder.

[1] showed that the only operator to combine *any arbitrary* preference relations that satisfies the desired properties IBUT (independence of irrelevant alternatives, based on preferences only, unanimity with abstentions, and preservation of transitivity) is the priority operator, which assumes that priority is a partial order. We observe here that if only *Boolean* preference relations (such as those resulting from Boolean simple criteria) are combined, the cardinality-based rule, in which all combined relations have equal priority, also satisfies the properties IBUT. Requiring antisymmetry in this case would unnecessarily restrict the expressivity.

**Acknowledgements** This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs. It is part of the Pocket Negotiator project with grant number VICI-project 08075. It is also partially supported by the New Governance Models for Next Generation Infrastructures project with NGI grant number 04.17.

## References

- [1] H. Andréka, M. Ryan, and P.-Y. Schobbens. Operators and laws for combining preference relations. *J. Log. Comput.*, 12(1):13–53, 2002.
- [2] C. Boutlier, R.I. Brafman, C. Domshlak, H.H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.*, 21:135–191, 2004.
- [3] G. Brewka. A rank based description language for qualitative preferences. In *Proc. ECAI*, p. 303–307, 2004.
- [4] W. Visser, R. Aydoğan, K.V. Hindriks, and C.M. Jonker. A framework for qualitative multi-criteria preferences. In *Proc. ICAART*, p. 243–248, 2012.

# Ordered Epistemic Logic: Semantics, Complexity and Applications

Hanne Vlaeminck    Joost Vennekens    Maurice Bruynooghe    Marc Denecker

*Department of Computer Science, KULeuven, Belgium*

## Abstract

Summary of a paper appearing in Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012. AAAI Press 2012.

Ordered Epistemic Logic (OEL) was first defined under the name *Hierarchical Autoepistemic Theories* by Konolige [10]. He observed that, in non-monotonic reasoning, the notion of inference from a *specific* body of knowledge often plays an important role. Recently, OEL was independently reintroduced by [8] in order to merge the contributions of ASP [5, 1] on the level of Knowledge Representation into classical first order logic (FO). As many of the original motivating examples of ASP involve defaults and autoepistemic propositions [9], this was done by adding an epistemic operator to FO.

We observed that these motivating examples for ASP, as well as many examples of Default Logic (DL) [13], often have a simple stratified structure: the goal is to reason on an existing incomplete knowledge base, e.g., by adding default assumptions. More complex examples can have several levels of stratification. However, neither autoepistemic logic (AEL) [11], default logic, nor ASP preserve this inherent stratification. As a result, a theory in each of these logics is a theory that refers to its own information content through a reflexive epistemic operator (see [6] for a recent account). This is a source of complexity that complicates both their semantics and their reasoning procedures.

By contrast, OEL maintains a stratified representation where each level extends the knowledge of the lower levels. This simplifies the logic considerably, while still being able to handle a lot of useful applications from AEL or DL, as shown in the full paper. Contrary to AEL, DL or ASP, an OEL theory always defines a unique belief set, represented as a set of possible worlds. The full paper shows that OEL solves some well-known problems of ASP in the context of epistemic applications. Syntactically, OEL extends FO; the only difference with FO is that OEL is a *closed domain* version of FO: all possible worlds share the same domain and interpretation of terms. This is like in many first order modal logics. With exception of this feature, OEL is a conservative extension of FO; its epistemic operator stands orthogonal to many other extensions of FO (e.g., types, inductive definitions, aggregates,...), and hence seamlessly integrates with them. By combining them, a very rich KR language is obtained in which many of the motivating examples in DL, AEL and ASP, as well as other extensions of FO such as FO(ID) [7], have a natural expression.

We here extend the initial work in [10, 8], in several ways. First, we prove that, in a given finite domain, the data complexity of model checking, satisfiability checking and query answering for OEL theories is in  $\Delta_2^P$ , which is indeed lower than for AEL and DL, where some instances of satisfiability checking problems can be proven to be  $\Sigma_2^P$ -complete. We also show how a model generator for OEL can be implemented.

Second, we illustrate the use of OEL and of model generation in the context of a scheduling problem with an epistemic component. Third, we extend OEL to a logic for distributed epistemic agents, which we call *distributed ordered epistemic logic* (d-OEL). Knowledge bases are still hierarchically ordered, but now theories at one level no longer automatically possess all the knowledge of lower levels.

Distributed ordered epistemic logic can cope with distributed knowledge, which makes it relevant for a number of new application areas. One example is the specification of access control policies. Formal specification languages used for this purpose (e.g., [2]) often include a construct allowing one policy manager to query the knowledge of another. Another potential application area is the Semantic Web, which can be seen as a huge network linking different sources of data and knowledge, often in the form of ontologies. Several proposals have been made to combine data from such ontologies, for example, through the use of *bridge*

rules [4]. Others address the need for expressing defaults such as “if source  $x$  does not specify the color of the car, we assume its color is black”. In yet other approaches, the web is seen as an open environment consisting of purely positive knowledge bases that do not contain negative information. This is to avoid potential inconsistencies between different sources [3]. Here, a limited form of “negation-as-failure” called *scoped negation* [12, 3] has been proposed: this is an epistemic operator to query whether a specific source (e.g., some ontology) does not know a proposition.

Common to all such applications is the presence of a number of knowledge sources that can query each other through some form of epistemic operator. We argue in the full paper that the logic d-OEL provides a simple formalism with a precise semantics to tackle some of these applications in a natural way.

The full paper starts with recalling some preliminaries. Next, syntax and semantics of both OEL and d-OEL are defined, examples are given and a number of properties about both languages are proved. The paper continues with a closer look at model generation for OEL and d-OEL and with proving our complexity results. After that, related work is discussed in detail, in particular the relationship with Answer Set programming, with Default Logic and with other hierarchical approaches but also with other formalisms for handling distributed (possibly contradictory) knowledge. The paper ends with a brief discussion.

## References

- [1] C. Baral. Knowledge representation, reasoning and declarative problem solving. 2003.
- [2] S. Barker. The next 700 access control models or a unifying meta-model? In *Proceedings of the 14th ACM symposium on Access control models and technologies, SACMAT '09*, pages 187–196. ACM, 2009.
- [3] T. Berners-Lee, E. P. Dan Connolly, and Y. Scharf. Experience with N3 rules. In *W3C Workshop on Rule Languages for Interoperability*, 2005.
- [4] P. Bouquet, F. Giunchiglia, F. V. Harmelen, L. Serafini, and H. Stuckenschmidt. C-OWL: Contextualizing ontologies. In *Journal Of Web Semantics*, pages 164–179. Springer Verlag, 2003.
- [5] G. Brewka, T. Eiter, and M. Truszczyński. Answer set programming at a glance. *CACM*, 54(12):92–103, 2011.
- [6] M. Denecker, V. W. Marek, and M. Truszczyński. Reiter’s default logic is a logic of autoepistemic reasoning and a good one, too. *CoRR*, abs/1108.3278, 2011.
- [7] M. Denecker and E. Ternovska. A logic of nonmonotone inductive definitions. *ACM Transactions on Computational Logic (TOCL)*, 9(2):Article 14, 2008.
- [8] M. Denecker, J. Vennekens, H. Vlaeminck, J. Wittocx, and M. Bruynooghe. Answer set programming’s contributions to classical logic. An analysis of ASP methodology. In M. Balduccini and S. Tran, editors, *MG-65: Symposium on Constructive Mathematics in Computer Science*. Lexington, October 26-27 2010.
- [9] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.
- [10] K. Konolige. Hierarchic autoepistemic theories for nonmonotonic reasoning. In *AAAI*, pages 439–443, 1988.
- [11] R. C. Moore. Possible-world semantics for autoepistemic logic. In *Proceedings of the Workshop on Non-Monotonic Reasoning*, pages 344–354, 1984. Reprinted in: M. Ginsberg, ed., *Readings on Nonmonotonic Reasoning*, pages 137–142, Morgan Kaufmann, 1990.
- [12] A. Polleres, C. Feier, and A. Harth. Rules with contextually scoped negation. In *Proc. 3rd European Semantic Web Conf. (ESWC2006)*, pages 332–347. Springer, 2006.
- [13] R. Reiter. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132, 1980.

# An efficiently computable support measure for frequent subgraph pattern mining

Yuyi Wang<sup>a</sup>

Jan Ramon<sup>a</sup>

<sup>a</sup> *Department of Computer Science, K. U. Leuven*

Frequent subgraph pattern mining is an important graph mining task. While the majority of existing systems focus on the transactional setting, where every transaction is a separate, independent graph, in recent years there has been an increasing interest in mining large networks. In particular, given a network (also called database graph)  $D$ , a pattern language  $L$ , a frequency measure  $freq$  and a minimal frequency threshold  $minsup$ , the task of frequent pattern mining is to list all patterns  $P \in L$  such that  $freq(D, P) \geq minsup$ . Unfortunately, in this problem formulation choosing a good frequency measure  $freq$  has shown to be challenging [1, 2, 3, 4]. Ideally, a frequency measure (to measure the number of occurrences of a pattern  $P$  in a graph  $D$ ) is a function satisfying the following properties:

1. *Anti-monotonic* the support of a pattern should not be larger than the support of any subpatterns. Therefore, we cannot just use the number of images (an image is a subgraph of the database graph, and it is isomorphic to the pattern) of a pattern as its support.

2. *Normalized* if for every pattern which has only independent images in a database graph, its support in that database graph equals the number of images. Independent images mean that they do not overlap according to some notion of overlap, such as sharing a vertex or an edge. In this paper<sup>1</sup>, we use vertex-overlap.

3. *Statistical soundness* the function should give a measure of the number of independent observations of a phenomenon (the pattern).

An important class of anti-monotonic normalized support measures relies on *overlap graphs*. Given a database graph  $D$  and a subgraph pattern  $P$ , the vertices in the overlap graph  $G_P^D$  are the images of  $P$  in  $D$ , and two vertices are adjacent iff the images overlap in  $D$  (here, non-overlap is used as an approximation of statistical independence). An overlap graph based support measure (OGSM) takes an overlap graph of a pattern in a database graph as its input, and outputs the support of that pattern in that database graph. Vanetik et al. [1] proposed the first OGSM, the size of the maximum independent set (MIS) of the overlap graph. Unfortunately, computing the MIS of an overlap graph is NP-hard, and it has been shown that MIS cannot be approximated even within a factor of  $n^{1-o(1)}$  efficiently [5], where  $n$  is the order of the overlap graph. Calders et al. [4] proposed the Lovász  $\vartheta$  value (see, e.g., [6, 7]), which is computable in time polynomial in the order of the overlap graph using semidefinite programming (SDP). A straightforward application of a general purpose SDP solver yields a running time of  $O(n^{6.5})$ . Approximation algorithms exist but even these approximative methods are still computationally too expensive for our purposes.

In this paper, we propose a new support measure  $s$  which is a solution to a (usually sparse) linear program.

As we are using vertex-overlap, each vertex  $v$  in a database graph  $D$  determines a clique in the overlap graph  $G_P^D$  in which  $P$  is a pattern. That is, the images which share the vertex  $v$  build a clique in  $G_P^D$ . Based on this observation, we introduce the overlap hypergraph  $H_P^D$  whose vertices are the images of  $P$  in  $D$ , and its hyperedges are these cliques. The  $s$  measure is an overlap hypergraph based support measure (OHSM). In order to define the  $s$  measure, a vector  $x$  indexed by the vertices of  $H_P^D$  is assigned to the  $H_P^D$ , i.e., the variable  $x_v$  is assigned to the vertex  $v$ . The linear program of the  $s$  measure is,

$$s(H_P^D) = \max_x \left\{ \sum_{v \in V(H_P^D)} x_v \mid \forall v \in V(H_P^D) : x_v \geq 0 \text{ and } \forall e \in E(H_P^D) : \sum_{v \in e} x_v \leq 1 \right\}. \quad (1)$$

<sup>1</sup>A longer version has been published in [12].

There are very effective methods for solving (1), including the simplex method, and the more recent interior-point methods [8]. The simplex method is efficient in practice though its complexity is exponential, and the interior-point method solves a linear program in  $O(n^2m)$ , where  $n$  is the number of variables (here,  $\min\{|V(H_P^D)|, |E(H_P^D)|\}$ , note that we can solve the dual program if necessary) and  $m$  is the number of constraints (here,  $|V(H_P^D)| + |E(H_P^D)|$ ). In addition, the size of the pattern  $P$  is usually small, so the linear program is often sparse, and most LP solvers can solve a sparse problem more efficiently.

We prove that the  $s$  measure is anti-monotonic and normalized. For this purpose, we provide the necessary and sufficient conditions of anti-monotonicity for any OHSM. Using these conditions, we also show that all normalized anti-monotonic OHSMs are bounded. Besides, compared to the min-image based support measure [2] which is not overlap based, the  $s$  measure has statistical advantages.

We perform experiments to verify the effectiveness and efficiency of our  $s$  measure.

In the first experiment, we compare the computational efficiency between our  $s$  measure and the Lovász  $\vartheta$  measure. We generate hypergraphs (20, 40, . . . , 200 vertices and 20, 40, . . . , 100 hyperedges) randomly, and convert them into graphs by replacing the hyperedges with cliques. The hypergraphs are used to compute the  $s$  measure, while the graphs are used to compute the Lovász  $\vartheta$  measure. For all the randomly generated (hyper)graphs,  $s$  can be computed in a very short period of time ( $< 0.01$  seconds), while the time consumed to compute  $\vartheta$  grows fast when the number of vertices increases. We can rationally claim that for larger (hyper)graphs on which the  $s$  measure can be computed efficiently, it is extremely difficult to compute the  $\vartheta$  value in a reasonable time period by solving the corresponding SDP using existing methods. Therefore, the  $s$  measure outperforms the  $\vartheta$  value in terms of efficiency.

Then, we mine frequent patterns of level up to 6 in real-world networks (DBLP0305, DBLP0507 [9]) and in synthetic scale-free networks of different sizes [11]. In these experiments, VF2 [10] is used to find all embeddings. Using VF2 and the  $s$  measure, frequent patterns of level up to 6 can be mined in a reasonable amount of time. In contrast to earlier approaches using the MIS or the  $\vartheta$  measures, the bottleneck in our experiments is the pattern matching part of the algorithm. On the real-world data, the time needed to compute embeddings is significantly larger than the time needed to compute  $s$ . For the larger synthetic datasets and the larger patterns the difference is even several orders of magnitude. If this part can be improved, it can be expected that larger patterns can be mined in larger networks.

## References

- [1] Vanetik N., Shimony S.E., and Gudes E.: Support measures for graph data. *Data Min. Knowl. Discov.* 13(2), 243–260 (2006)
- [2] Bringmann B., Nijssen S.: What is frequent in a single graph? *Proc. of PAKDD'08*, 858–863 (2008)
- [3] Fiedler M., Borgelt C.: Support Computation for Mining Frequent Subgraphs in a Single Graph. *Proc. of MLG'07*, (2007)
- [4] Calders T., Ramon J., Dyck D.V.: All normalized anti-monotonic overlap graph measures are bounded. *Data Min. Knowl. Discov.* 23(3), 503–548 (2011)
- [5] Feige U., Goldwasser S., Lovász L., Safra S., Szegedy M.: Approximating clique is almost NP-Complete. *FOCS*, IEEE Computer Society, 2–12 (1991)
- [6] Lovász L.: On the Shannon capacity of a graph. *IEEE Trans. on Info. Theory* 25(1), 1–7 (1979)
- [7] Knuth D.E.: The sandwich theorem. *Electr. J. Comb.* 1, 1–48 (1994)
- [8] Boyd S., Vandenberghe L.: *Convex optimization*. Cambridge Univ. Press (2004)
- [9] Berlingerio M., Bonchi F., Bringmann B., Gionis A.: Mining graph evolution rules. *Proc of ECML/PKDD'09*, 115–130 (2009)
- [10] Luigi P.C., Pasquale F., Carlo S., Mario V.: A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(10): 1367–1372 (2004)
- [11] Barabási A.L., Albert R.: Emergence of scaling in random networks. *Science*, 286, 509–512 (1999)
- [12] Wang Y., Ramon J.: An efficiently computable support measure for frequent sungraph pattern mining. *Proc. of ECML/PKDD'12*, 362–377 (2012)

# Enhancing predictability of schedules by task grouping<sup>1</sup>

Michel Wilson<sup>a</sup>

Cees Witteveen<sup>a</sup>

Bob Huisman<sup>b</sup>

<sup>a</sup>*Delft University of Technology, Dept of Software Technology*

<sup>b</sup>*Nedtrain, Fleet Services – Maintenance Development*

In a scheduling problem one has to determine time slots for a set of tasks (activities) to be completed, subject to time and resource constraints. An important problem in applying scheduling methods is the *predictability* of scheduling solutions: when encountering task execution delays, one would want to minimize recomputations and changes of the currently constructed schedule as much as possible. Two possible approaches to ensure predictability can be distinguished [3]: *on-line* reactive approaches and *off-line* or pro-active approaches.

Our paper pursues the latter approach, by coming up with a set  $S$  of schedules instead of just one schedule, such that, during execution time, the currently chosen schedule  $\sigma \in S$  can be replaced by another one  $\sigma' \in S$  such that it meets the changed problem constraints without affecting the objectives. The method is based on a transformation of the original scheduling problem, by *grouping* some tasks together into a new composite task. By grouping, the task ordering within a group is left unspecified in creating a schedule. Only at execution time a specific, most suitable, ordering of the original tasks making up the composite tasks is decided. This method increases the make span of the schedule, but as a result the predictability in the face of uncertainty during execution is increased as well.

## 1 Task grouping

The procedure of task grouping is based on the Precedence Constraint Posting approach [1] to solving the Resource Constrained Project Scheduling Problem (RCPSP). The basic precedence constraint procedure iterates over four steps: 1) compute the resource usage profile over time, using an earliest start time solution, and select a peak, i.e., a point at which the use of a resource exceeds its capacity, 2) select two (partially) concurrent tasks contributing to the selected peak, 3) decide on a sequential order for these two tasks, and add a precedence constraint to the problem enforcing this order, and finally 4) calculate updated earliest start times to create a new schedule.

Our task grouping approach involves a simple modification to the procedure above, yielding more execution-time flexibility: instead of deciding on a *fixed ordering* of the two selected tasks in step two, we remove the two tasks from the problem, and replace them by a grouped task, which reserves enough resources to enable *any execution order* of the two tasks. The grouped task can be treated as any other task, in particular, it is possible for such a task to participate in another grouping operation, such that a group task can represent a reservation of resources for any number of tasks, which can be executed in any (sequential) ordering.

To control task grouping, the heuristic used to choose the constraint direction in regular constraint posting is used. Normally, the constraint between two tasks is posted in the direction preserving the largest amount of slack between the two tasks. In our algorithm, the two tasks are grouped if the *difference* between the slack of both orderings is below a threshold parameter  $\gamma$ .

## 2 Experimental results

As experiment, simulated executions were performed of schedules for the well-known benchmark instances from PSPLIB [2]. In each execution, delays were inserted in some of the tasks, and to estimate the

---

<sup>1</sup>This is an abstract of our paper accepted for the 2012 European Conference on Artificial Intelligence (ECAI)

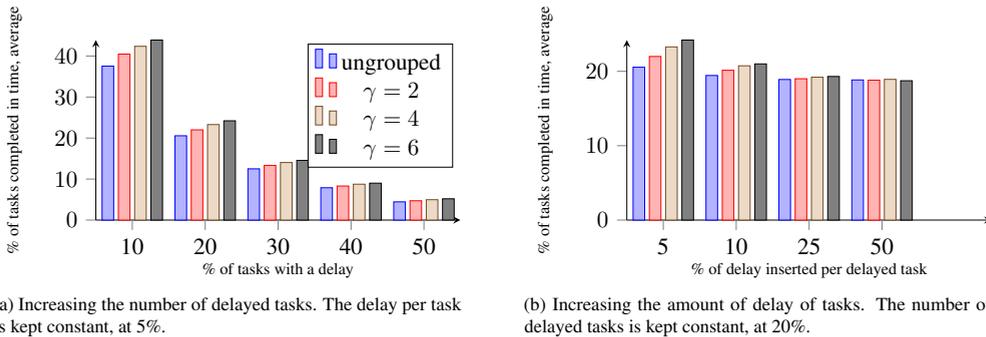


Figure 1: Schedule predictability for various values of  $\gamma$ , with increasing number of delayed tasks and amount of delay. All instances of the sets j30 and j60 were simulated, and the average percentage of tasks completed at their scheduled time is shown.

predictability of the schedule it was determined how many of the tasks were completed at their expected time, according to the original schedule. Additionally, the effect on the makespan of the schedules was measured.

A makespan increase is expected due to the worst case envelope of a group task, for both start times and resource usage. The experiments indeed show an increase, from 7.2% for instances with 30 tasks to 16.5% for instances with 120 tasks can be seen for  $\gamma = 2$ . For  $\gamma = 8$  the increase is bigger, from 19.8% to 43.1%.

For the predictability, two series of tests were performed: one in which the number of delayed tasks was varied, and one in which the amount of delay in the tasks was varied. The results are presented in Figures 1a and 1b. The number of tasks which complete in time drops rapidly when increasing the number of delayed tasks, which is in line with expectations. The performance of grouping increases for larger values of  $\gamma$ ; the cause is the larger number of groups present in the solution. If the number of delays gets large however, the gains of grouping diminish.

The amount of delay per task has a much lower effect on the predictability: longer delays do not cause additional tasks to be completed late. This shows that the schedules do not contain a lot of slack: a small delay is already enough to propagate to all tasks following the delayed tasks. Here, it can be seen that our grouping method performs well for small amounts of delay. This is in line with expectations: if a task is delayed by a large amount, a reordering of the tasks succeeding it can rarely absorb the delay.

### 3 Conclusions

In this paper, a novel way to create predictable schedules is presented, using grouped tasks to enable some tasks to be re-ordered during execution time, to prevent the propagation of a delay. Additionally, the executing agent gains some autonomy during execution.

Tests using simulated execution show that schedules containing grouped tasks are indeed more predictable: more tasks are completed at their scheduled time when delays are inserted. The method works best for small delays, but it still works somewhat even if a large amount of the tasks incurs a small delay.

### References

- [1] Amedeo Cesta, Angelo Oddi, and Stephen F Smith. Profile-based algorithms to solve multiple capacitated metric scheduling problems. In *Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems*, pages 214–223, 1998.
- [2] Rainer Kolisch, Christoph Schwindt, and Arno Sprecher. Benchmark instances for project scheduling problems. In *Handbook on Recent Advances in Project Scheduling*, pages 197–212. Kluwer, 1998.
- [3] Nicola Policella, Amedeo Cesta, Angelo Oddi, and Stephen Smith. Solve-and-robustify: Synthesizing partial order schedules by chaining. *Journal of Scheduling*, 12:299–314, 2009. 10.1007/s10951-008-0091-7.

# Mechanism for Robust Procurements<sup>1</sup>

Yingqian Zhang <sup>a</sup>

Sicco Verwer <sup>b</sup>

<sup>a</sup> *Erasmus University Rotterdam, The Netherlands*

<sup>b</sup> *Radboud University Nijmegen, The Netherlands*

The increasing popularity of adopting auctions is largely due to its efficiency of allocating goods. However, in the face of uncertainties on services, the winner determination solution is often not robust enough to ensure a reliable outcome. This paper aims to design a more robust auction by introducing redundancy into the selected solution. More specifically, we construct an algorithm and a mechanism for incentivizing truth-telling in public procurement problems with uncertainties. Our contributions are the development of a framework for studying such procurement problems, proving that minimizing cost in this framework is NP-complete, developing a quick algorithm that minimizes this cost, and providing a novel multi-stage mechanism that has desirable properties such as efficient, truthful in dominant strategies, and post-execution individually rational. We show experimentally that our approach significantly outperforms the current practice in many settings.

**Robust procurement problem (RPP)** The procurer announces a job with a deadline  $D$  and a minimal completion probability  $\gamma$ . Each bidder  $i \in A$  submits a bid that consists of  $\langle c_i, d_i, \beta_i \rangle$ , where  $c_i$  is the cost of executing the job,  $d_i$  is the duration of completing the job, and  $\beta_i$  specifies the reservation fee. We assume the auctioneer has the information about the reliability  $r_i \in [0, 1]$  of each participant bidder  $i$ . Given the set of bids and the reliability of agents, the procurer determines a set of winners  $S = (A_{i_1}, \dots, A_{i_m}) \subseteq A$  as the outcome  $\phi$ .  $S$  is a total ordered set. The probability that an ordered set of contractors  $S = (A_{i_1}, \dots, A_{i_m})$  will finish the project within the deadline  $\sum_{k=1}^m d_{i_k}$  is equal to  $1 - \prod_{k=1}^m (1 - r_{i_k})$ . The expected cost incurred by  $S$  then becomes:  $E[Cost(S)] = \sum_{k=1}^m (c_{i_k} + \beta_{i_{k+1}}) \prod_{l=0}^{k-1} (1 - r_{i_l})$ , where  $r_{i_0} = 0$  and  $\beta_{i_{m+1}} = 0$ . Denote by  $\mathbb{S}$  the possible ordered sets of contractors that may finish the project within deadline  $D$  with probability at least  $\gamma$ , that is:  $\mathbb{S} = \{(A_{i_1}, \dots, A_{i_m}) : \sum_{k=1}^m d_{i_k} \leq D \text{ and } \prod_{k=1}^m (1 - r_{i_k}) \leq 1 - \gamma\}$ . The robust procurement problem (RPP) can be now defined as the following constrained optimization problem:  $\min_{S \in \mathbb{S}} E[Cost(S)]$ .

We consider a setting where agents are self-interested and their declarations are private information. Let this so-called *type* of each agent  $i$  be denoted by  $\theta_i$ . We use  $\theta_{-i}$  to denote the type profile without the type of agent  $i$ . Given a type profile, a direct-revelation mechanism selects an *outcome*  $\phi = f(\theta)$  using an algorithm  $f$  from the set of possible outcomes, and an expected payment  $\bar{p}_i(\phi, \theta)$  for each agent that together define the expected utility of an agent  $\bar{u}_i(\phi, \theta) = \bar{v}_i(\phi, \theta_i) - \bar{p}_i(\phi, \theta)$ .  $\bar{v}_i(\phi, \theta_i)$  specifies the expected valuation of agent  $i$  on the outcome  $\phi$ . Let  $S_\phi = (A_{a_1}, \dots, A_{a_m})$  denote the ordered set given the outcome  $\phi$ . The expected valuation of agent  $i$  (i.e.  $A_{a_i}$ ) prior to execution is computed as:  $\bar{v}_i(\phi, \theta_i) = -c_i \prod_{l=0}^{i-1} (1 - r_l) - \beta_i \prod_{l=0}^{i-2} (1 - r_l)$ . Note the realized valuation  $v_i(\phi, \theta_i)$  of a contractor  $A_{a_i}$  after execution of the procurement schedule depends on the actual execution. The expected *social welfare* is defined as:  $\bar{w}(\phi) = \sum_{i=1}^m \bar{v}_i(\phi, \theta_i)$ . The mechanism is called *efficient* when it selects outcomes where the expected social welfare is highest. In addition, we are interested in mechanisms that are truthful in dominant strategies and *post-execution* individually rational, i.e., a truthful agent will not receive a negative utility no matter what the actual execution outcome will be.

**Complexity and algorithm for RPP** We show that the robust procurement problem RPP is computationally hard by showing that the subproblem (i.e., the RPP without costs problem) of finding a suitable set of contractors is NP-complete. The proof is by reduction from the subset sum problem. Although this means that it is difficult to optimize in theory, in practice an optimal solution can often be found efficiently because

<sup>1</sup>The full version of the paper appears in I. Rahwan et al. (Eds.): PRIMA 2012: Principles and Practice of Multi-Agent Systems. LNAI 7455, pp. 77–91. Springer, 2012.

the number of contractors with a total duration less than the job deadline is typically small. Furthermore, the number of contractors required to reach the desired reliability is usually limited. The total search space is therefore typically small, and even a brute-force search should often be able to find the optimal solution within reasonable time. We propose a backtracking algorithm that iteratively appends one contractor to a partially constructed solution. The algorithm uses bounds to further reduce the search space: (1) Adding a contractor to the end of an ordering always increases the expected cost of the ordering. (2) Adding a contractor always increases the total duration of an ordering.

**Mechanism for RPP** One possible truthful mechanism is the VCG mechanism. However, it has been shown in [1] that VCG is individual rational in expectation only, i.e., an agent may get a negative (realized) utility after the execution of the procurement outcome. In this paper, we define a multi-stage Grove mechanism and show that it is efficient, truthful in dominant strategies, and post-execution individually rational. The proposed mechanism, called RobustProcurement, works as follows:

The auctioneer announces a job with deadline and completion probability threshold  $\gamma$ .

1. The contractor agents declare their types  $\theta = (\theta_i, \theta_{-i})$  to the auctioneer.
2. The auctioneer then finds an optimal schedule  $\phi$  using the proposed algorithm.
3. Every agent  $i$  in the ordered set  $S_\phi = (A_{a_1}, \dots, A_{a_m})$  receives its marginal contribution as payment:  $p_i(\phi, \theta_i) = -\bar{w}(\phi, \theta) + \bar{w}_{-i}(\phi', \theta_{-i})$ , where  $\bar{w}(\phi, \theta)$  is the expected social welfare,  $\phi' = f(\theta_{-i})$  is the efficient outcome without agent  $i$ 's participation, and  $\bar{w}_{-i}(\phi', \theta)$  is the social welfare on  $\phi'$ .
4. The first winner  $A_{a_1}$  receives an additional payment:  $p'_i(\phi, \theta_i) = v_i(\phi, \theta_i)$ .  $A_{a_1}$  starts to execute the job. The second winner  $A_{a_2}$  is notified to be stand-by. The reservation cost is incurred.
5. At the deadline of agent  $A_{a_{i-1}}$  for  $2 \leq i \leq |S| - 1$ , do
  - Transfer the additional payment to agent  $i$  (i.e. agent  $A_{a_i}$ ), where the agent's realized valuation  $v_i(\phi, \theta_i)$  is computed based on the realization of the schedule  $\phi$ , i.e.,

$$v_i(\phi, \theta_i) = \begin{cases} -c_i - \beta_i & \text{if } A_{a_{i-1}} \text{ does not complete the job by its deadline;} \\ -\beta_i & \text{otherwise.} \end{cases}$$

- If  $A_{a_{i-1}}$  completes the job, inform the remaining agents  $j \in S$ , and terminate the mechanism. Otherwise, agent  $A_{a_i}$  starts to execute the job, and agent  $A_{a_{i+1}}$  is notified to be stand-by.
- Loop step 5 until the mechanism is terminated.

**Experiments** We investigate the performance of our mechanism and compare it to a greedy mechanism that represents the current practice in procurement: create a first-price auction, select a winning contractor, and create a new auction if the contractor fails. The mechanism iteratively selects a contractor. The performance measurements are: (1) the expected social welfare (i.e., the total cost incurred by the contractors) and (2) the expected payments of the auctioneer. We create many different sets of agents for a given combination of deadlines and reliability thresholds, and the number of agents. For every combination, we generate 50 problem instances. Every instance is solved using our robust procurement algorithm and compared with the result of running the aforementioned greedy procedure using three different costs for every iteration after the first. We compute a greedy solution first for 0 re-iteration costs, then one for 1 time the average reservation costs of the agents, and finally 2 times the average reservation costs.

The results show that in terms of social welfare, our mechanism outperforms the greedy approach in all cases except when there exist cheap and reliable agents who can finish the job in time. In terms of payments, our mechanism outperforms the current practice when there are many potential contractors and constraints in the optimization problem are tight. The results are promising especially considering that in the experiments, the potential cost increase due to misreporting of the agents is disregarded in the greedy mechanism.

## References

- [1] Sebastian Stein, Enrico H. Gerding, Alex Rogers, Kate Larson, and Nicholas R. Jennings. Algorithms and mechanisms for procuring services with uncertain durations using redundancy. *Artif. Intell.*, 175(14-15):2021–2060, 2011.

# Agent Programming Languages Requirements for Programming Cognitive Robots

Pouyan Ziafati <sup>ac</sup>

Mehdi Dastani <sup>c</sup>

John-Jules Meyer <sup>c</sup>

Leendert van der Torre <sup>ab</sup>

<sup>a</sup> *SnT, University of Luxembourg*

<sup>b</sup> *CSC, University of Luxembourg*

<sup>c</sup> *Intelligent Systems Group, Utrecht University*

## Abstract

This paper<sup>1</sup> presents various requirements for BDI-based agent programming languages to provide better support for implementing autonomous robotic control systems. Examples of such requirements are: 1- Built-in support for integration with existing robotic frameworks such as ROS<sup>2</sup>, 2- Real-time reactivity to events, 3- Management of heterogeneous sensory data and reasoning on complex events, and 4- Representation of complex plans and coordination of the parallel execution of plans.

## 1 Introduction

To achieve complex goals in dynamic environments, robots need to be empowered with a deliberative behavior. One of the suitable architectures for implementing deliberative behavior is the BDI architecture [4]. Various agent programming languages such as 2APL [1] have been designed and developed to facilitate the implementation of BDI architecture. However, the application domains of these languages have been mainly limited to cognitive software agents.

Our research aim is to provide necessary requirements to facilitate the use of BDI-based agent programming languages for implementing robotic control systems in a modular and systematic way. This paper presents various requirements that agent programming languages should address to be more suitable for programming cognitive robots. The requirements investigated are related to the problems of plan execution control, sensory data management and real-time reactivity. We faced these problems as basics when exploring our home-care NAO service robot application scenario presented in the original paper<sup>1</sup>.

## 2 Requirements for Programming Cognitive Robots

Studying the problem of programming cognitive robots and looking into current robotic architectures and tools [3, 2, 5] shows that current agent programming languages lack support for different aspects of programming cognitive robots. Our aim is to enable such support for BDI-based agent programming languages in general and 2APL agent programming language in particular.

One of such requirements is integration with existing robotic frameworks. This can encourage the use of agent programming languages by robotic community and facilitate their use for rapid prototyping and development of autonomous systems. To address this requirement, we have developed an environment interface for 2APL facilitating its integration with the ROS framework. We have used ROS to provide basic robotic capabilities such as face recognition, voice recognition and a number of high-level actions such as

---

<sup>1</sup>This compressed contribution is based on “Ziafati, P. Dastani, M. Meyer, J. J. Van Der Torre, L. Agent Programming Languages Requirements for Programming Cognitive Robots, Proceedings of the Tenth International Workshop on Programming Multi-Agent Systems, ProMAS @ AAMAS 2012, Pages 39-54”.

<sup>2</sup><http://www.ros.org/wiki/>

stand-up(), turn-neck(O) and walk-to(X,Y) for our NAO robots. Using 2APL and ROS, we have developed a demo application in which different NAO's movement can be controlled by voice. Also NAO can remember faces and whenever a user greets NAO, if NAO recognizes the user's face, it greets the user by his/her name.

The second requirement is providing support for the development of a sensory component that processes and manages heterogeneous sensory information. This component should enable unified representation of sensory data and domain knowledge as well as reasoning on high-level events (i.e. situations). The sensory information managed and processed by the sensory component should be accessible by a BDI-based control component in a symbolic form and through both querying and receiving of events.

Another requirement is extending plan representation and execution capabilities of current agent programming languages. Required capabilities include providing support for governing the execution of plans by sequential, temporal and priority orderings, and based on different internal conditions and external events, representing and handling conflicts in parallel execution of plans, and monitoring and handling plans execution failures.

The last requirement is real-time reactivity to events. Based on the current trend in design of the state of the art robotic control architectures [2, 5], a suitable approach for addressing such requirement is providing support for development of distributed real-time BDI-based control systems. This requires a specific version of an agent programming language dedicated to development of real-time control components. The semantic and implementation of such a version should guarantee safe and bounded-time computations to enable analysis and guaranteeing required real-time properties of the control component. Also a dedicated architecture and runtime environment is required to support the real-time coordination and communication of different control components of a robot. We envision an architecture consisting of a distributed set of BDI-based control components with different functionalities (e.g. deliberative, reactive, plan failure handling) which can share beliefs and goals, and other robotic software components including sensory components described above. These components can have different real-time requirements. Some of them should be run in real-time and guarantee bounded reaction and response time to events.

### 3 Conclusion

The paper presents various requirements for agent programming languages in order to provide better support for implementing cognitive robotic control systems. These requirements are drawn partially based on an analysis of the problem at hand and partially based on a study of current robotic development tools. We do not claim that the set of requirements presented in this paper is complete. For example multi-robot scenarios impose other requirements for coordination, cooperation and communication between robots which have been left for further research. However our work contributes to the systematic analysis and presentation of agent programming languages requirement for programming cognitive robots.

### Acknowledgement

Pouyan Ziafati is supported by the National Research Fund (FNR), Luxembourg.

### References

- [1] M. Dastani. 2APL: a practical agent programming language, *Autonomous Agents and Multi-Agent Systems*, Volume 16 , Issue 3, Pages: 214 - 248, 2008.
- [2] R. Passama and D. Andreu. ContrACT: a software environment for developing control architecture. 6th National Conference on Control Architectures of Robots, 16 p, 2011
- [3] K. Rajan, et. el. Onboard Adaptive Control of AUVs using Automated Planning and Execution, Intl Symposium on Unmanned Untethered Submersible Technology (UUST), Durham, NH, August 2009.
- [4] A. Rao and M. Georgeff. BDI Agents: From Theory to Practice. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 312-319, June 1995.
- [5] V. Verma, et. el. Survey of command execution systems for NASA spacecraft and robots. In *Plan Execution: A Reality Check Workshop at the International Conference on Automated Planning and Scheduling (ICAPS)*, 2005

# A BDI Dialogue Agent for Social Support: Specification and Evaluation Method<sup>1</sup>

J.M. van der Zwaan      V. Dignum      C.M. Jonker

*Delft University of Technology, P.O.Box 5010 2600 GA Delft*

## Abstract

An important task for empathic agents is to provide social support, that is, to help alleviate emotional distress. In this paper we specify verbal support types for a dialogue agent that provides social support, and propose an evaluation method for such an agent in a sensitive domain (cyberbullying) with a vulnerable target audience (children).

## 1 Introduction

Social support or comforting refers to communicative attempts to alleviate emotional distress and is aimed at increasing the well-being of people and decreasing the perceived burden of their problems. Recent developments in affective computing show that empathic agents are increasingly capable of complex social and emotional dialogues. However, these dialogues are predominantly task oriented, i.e. to help the user perform a concrete task, such as finding information or learning. A comforting conversation is focussed on giving and receiving support; no concrete tasks are involved.

In our research, we are investigating how and to what extent Embodied Conversational Agents (ECAs) can provide social support. Recently, we proposed a dialogue model for social support [2]. Interaction between the agent and the user takes place in two main stages: 1) Gather information about the current situation, 2) Give advice on how to deal with the situation. This paper presents an extension of the dialogue model by specifying strategies for verbal social support that frequently occur in online counseling conversations (sympathy, compliment, encouragement, advice, and teaching).

The extended dialogue model was implemented in a conversational agent that provides social support to victims of cyberbullying. Given the sensitivity of the application domain (cyberbullying) and the vulnerability of the target audience (children), a careful and thorough evaluation is highly important. Additionally this paper presents an evaluation plan for the dialogue agent. While it is tempting to think about the agent in terms of its potential for reducing cyberbullying, we would like to emphasize that our focus is on providing social support.

## 2 Verbal Social Support

To illustrate how the agent verbally expresses social support, we specify the reasoning process behind sympathy. The agent's reasoning engine is modeled according to the Belief-Desire-Intention (BDI) paradigm [1]. This means the agent has beliefs (e.g., about what advice to give in which situations), goals (e.g., to give social support), and plans (e.g., to gather information about the upsetting situation). To improve the readability of the explanation, all speech acts have been translated into natural language utterances.

The information gathering phase of a comforting conversation consists of a recurring pattern of the agent asking a question, the user answering that question, and the agent acknowledging the answer. An acknowledgement is either neutral (e.g., *Okay*) or sympathetic. The agent expresses sympathy if it follows from his beliefs sympathy is applicable, otherwise it plays safe by staying neutral. Table 1 shows an example

---

<sup>1</sup>The full version of this paper has been accepted for publication at the workshop on Emotional and Empathic Agents (EEA) @AAMAS2012.

Example dialogue	Knowledge base
<b>Agent:</b> <i>Can you tell me what happened?</i> <b>User:</b> <i>Someone is calling me names on msn</i> <b>Agent:</b> <i>That's awful!</i> (sympathy)	<code>incident(type, call_names).</code> <code>incident(method, msn).</code>  <code>sympathy(type, call_names):-</code> <code>  incident(type, call_names).</code>

Table 1: Example dialogue fragment in which the agent expresses sympathy.

dialogue in which the agent expresses sympathy. The second column contains the contents of the agent's belief base. After the user answers the question two `incident` facts are added to the agent's belief base. The `sympathy` rule triggers the agent's sympathetic response. Absence of this rule would have resulted in a neutral acknowledgement of the user's input (e.g., *I see*). To enable other responses, more facts and rules have to be added to the belief base.

The other support types have been implemented in a similar manner. Like sympathy, compliment and encouragement occur in response to the answers the user gives to questions of the agent. Advice and teaching are uttered pro-actively, after the agent gathered sufficient information (what is sufficient depends on domain knowledge). For advice that requires an explanation, the agent optionally teaches the user how to execute the advice.

### 3 Evaluation Plan

This section describes the evaluation plan for the social support dialogue agent in the cyberbullying domain. The goal of the evaluation is to determine the extent to which users experience social support when interacting with the agent. The evaluation plan consists of multiple, incremental stages in which the dialogue system is improved based on the feedback from the previous stage before moving on to the next. In the first stage of the evaluation, we will perform an expert evaluation and create scenarios of common cyberbullying situations for indirect evaluation. After multiple experiments and incremental improvements on the dialogue agent we intend to involve children in the evaluation process. In the final stage of the evaluation, actual cyberbullying victims will be involved. Experiments in which children participate will be always conducted in cooperation with and under the supervision of psychologists and online counselors. Performance of the agent will be measured with questionnaires on perceived social support and trustworthiness of the agent.

### 4 Conclusion

In this paper, we specified the reasoning process of a dialogue agent that verbally expresses social support. Support types sympathy, compliment and encouragement are always given in response to user input. Advice and teaching are offered pro-actively. Additionally, we presented a multi-stage evaluation method for the dialogue agent. Because cyberbullying is a sensitive topic and children are a vulnerable target audience, we will start with an expert evaluation and create scenarios of common cyberbullying situations for indirect evaluation. After incremental improvements on the dialogue system, children and later cyberbullying victims will be involved in the evaluation process.

### Acknowledgements

This work is funded by NWO under the Responsible Innovation (RI) program via the project 'Empowering and Protecting Children and Adolescents Against Cyberbullying'.

### References

- [1] P.R. Cohen and H.J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213 – 261, 1990.
- [2] J.M. van der Zwaan, V. Dignum, and C.M. Jonker. A conversation model enabling intelligent agents to give emotional support. In *Studies in Computational Intelligence*, 2012.

**Demonstrations**

**BNAIC 2012**



# User-Computer Persuasion Dialogue for Grounded Semantics<sup>1</sup>

Martin Caminada <sup>a</sup>

Mikołaj Podlaszewski <sup>a</sup>

<sup>a</sup> *Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg*

We present an implementation <sup>2</sup> of the recently developed persuasion dialogue game for formal argumentation theory <sup>3</sup> under grounded semantics [3]. The idea is to apply Mackenzie-style dialogue [4, 5] to convince the user that an argument is or is not in the grounded extension. Hence, to provide a (semi-)natural user interface to formal argumentation theory.<sup>4</sup> Our approach is based on the concept of a complete labelling [2], which is a function that assigns each argument a label that is either *in*, *out* or *undec*, such that for each argument: (1) the argument is labelled *in* iff all its attackers are labelled *out* and (2) the argument is labelled *out* iff it has at least one attacker that is labelled *in*.

Standard argumentation theory states that an argument is in the grounded extension iff it is labelled *in* by each and every complete labelling [2]. Therefore, if one equates a complete labelling with a reasonable position an agent can take in the presence of the conflicting information represented by the argumentation framework, being accepted (labelled *in*) by every reasonable position (complete labelling) implies being able to convince even a maximally sceptical (but still reasonable) discussion partner. For this, we apply the concept of Mackenzie-style dialogue [4, 5]. Our theory differs from the Standard Grounded Game [6] in that: (1) we apply Mackenzie-style dialogue moves, like *claim*, *why*, *because* and *concede*, (2) when an argument is labelled *in*, we show that *all* its attackers are labelled *out* whereas in the Standard Grounded Game this is shown for only one of the attackers (at least in a single game or line of arguments), (3) we rely on the concept of a commitment store for determining the possible moves and ensuring correctness and completeness w.r.t. grounded semantics, (4) we do not apply the notion of a discussion tree, which after all is alien to Mackenzie-style dialogue, and (5) the presence of a winning strategy is not required to establish membership of the grounded extension; instead a single game won by the proponent against a maximally skeptical opponent is sufficient.

Our implementation uses a command-line interface, and is written in Python. The argumentation framework can either be loaded from a text file or entered manually. At the highest level, the user has eight commands at his disposal: *question*, *claim*, *load*, *save*, *af\_cat*, *af\_define*, and *quit*. With *question* the user asks the system about the status of a particular argument (say *A*). The system then responds either with *claim in(A)*, meaning that *A* has to be labelled *in* by every complete labelling (hence, *A* is in the grounded extension), with *claim out(A)*, meaning that *A* has to be labelled *out* by every complete extension (hence, *A* is attacked by the grounded extension) or with *no commitment A*, meaning that neither is the case. In the first two cases, the associated *claim* move is the start of a grounded dialogue as described in [3], which the user could choose to bypass by immediately conceding the main claim. When the user does a *claim* command, the system responds either by conceding (if it holds the claim that a particular argument has to be labelled *in* or *out* to be correct) or by holding a persuasion dialogue (if the system holds the claim to be incorrect). Although in the latter case, the discussion will in the end always be won by the system (since the ability to win the grounded persuasion game for a particular argument coincides with its membership of the grounded extension [3]) the discussion might still lead the user to valuable insight about why his initial position was wrong. With the *load*, *save*, *af\_cat* and *af\_define* commands one respectively loads, saves, displays or manually defines an argumentation framework. The dialogue game follows the

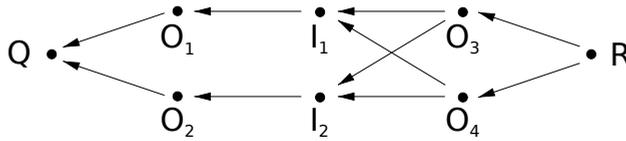
---

<sup>1</sup>Supported by the National Research Fund, Luxembourg (LAAMI and LAAMIcomp projects)

<sup>2</sup>The source code (GPL) and other necessary files can be downloaded at our project page <http://code.google.com/p/pyaf1/downloads/list>

<sup>3</sup>We refer to [1] for an overview of formal argumentation theory.

<sup>4</sup>This would for instance allow experimentation on user acceptance of formal argumentation results.



rules described in [3], with the exception that parties can terminate the dialogue at any point by conceding or withdrawing the main claim.

Assume the argumentation framework of the figure above. The interaction between the system and the user may look as follows.

Moves	Commitment			
	Proponent		Opponent	
	in	out	in	out
U: question $Q$	-	-	-	-
C: claim in( $Q$ )	$Q$	-	-	-
U: why in( $Q$ )	$Q$	-	-	-
C: because out( $O_1, O_2$ )	$Q$	$O_1, O_2$	-	-
U: why out( $O_1$ )	$Q$	$O_1, O_2$	-	-
C: because in( $I_1$ )	$Q, I_1$	$O_1, O_2$	-	-
U: why in( $I_1$ )	$Q, I_1$	$O_1, O_2$	-	-
C: because out( $O_3, O_4$ )	$Q, I_1$	$O_1, O_2, O_3, O_4$	-	-
U: why out( $O_3$ )	$Q, I_1$	$O_1, O_2, O_3, O_4$	-	-
C: because in( $R$ )	$Q, I_1, R$	$O_1, O_2, O_3, O_4$	-	-
U: concede in( $R$ ), out( $O_3, O_4$ ), in( $I_1$ ), out( $O_1$ )	$Q, I_1, R$	$O_1, O_2, O_3, O_4$	$R, I_1$	$O_3, O_4, O_1$
U: why out( $O_2$ )	$Q, I_1, R$	$O_1, O_2, O_3, O_4$	$R, I_1$	$O_3, O_4, O_1$
C: because in( $I_2$ )	$Q, I_1, R, I_2$	$O_1, O_2, O_3, O_4$	$R, I_1$	$O_3, O_4, O_1$
C: concede in( $I_2$ ), out( $O_2$ ), in( $Q$ )	$Q, I_1, R, I_2$	$O_1, O_2, O_3, O_4$	$R, I_1, I_2, Q$	$O_3, O_4, O_1, O_2$

Note that the argument  $O_4$  is directly conceded (without playing a why move), because the argument  $R$  was given as an answer to why out( $O_3$ ). In general one can notice that each argument appears in dialogue at most three times - once in a because (claim) move, at most once in a why move and once in a concede move - hence the length of the dialogue is linear in number of arguments. In contrast, applying the Standard Grounded Game [6] would require investigation of four lines:  $Q-O_1-I_1-O_3-R$ ,  $Q-O_1-I_1-O_4-R$ ,  $Q-O_2-I_2-O_3-R$ ,  $Q-O_2-I_2-O_4-R$ . Extending the example by duplicating four arguments  $I_1, I_2, O_3, O_4$  will double this number and in general case the number of lines of the Standard Grounded Game is exponential w.r.t the number of arguments.

Our plan is to keep developing it and integrate it with ArguLab [7]. Furthermore, we plan to implement a similar dialogue game for credulous preferred semantics.

## References

- [1] P. Baroni, M.W.A. Caminada, and M. Giacomin. An introduction to argumentation semantics. *Knowledge Engineering Review*, 26(4):365–410, 2011.
- [2] M.W.A. Caminada and D.M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009. Special issue: new ideas in argumentation theory.
- [3] M.W.A. Caminada and M. Podlaszewski. A persuasion dialogue for grounded semantics. In *In Proceedings of the Fourth International Conference on Computational Models of Argument (COMMA 2012)*, 2012. In print.
- [4] J. D. Mackenzie. Question-begging in non-cumulative systems. *Journal of Philosophical Logic*, 8:117–133, 1979.
- [5] J. D. Mackenzie. Four dialogue systems. *Studia Logica*, 51:567–583, 1990.
- [6] S. Modgil and M.W.A. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In I. Rahwan and G.R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–129. Springer, 2009.
- [7] M. Podlaszewski, Y. Wu, and M. Caminada. An implementation of basic argumentation components. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1307–1308. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

# COCALU: Convex outline collision avoidance under localization uncertainty [Demonstration]

Daniel Claes

Daniel Hennes

Karl Tuyls\*

*Department of Knowledge Engineering  
Maastricht University, P.O. Box 616, 6200 MD Maastricht*

\* Corresponding author: k.tuyls@maastrichtuniversity.nl

## 1 Introduction

Collision avoidance is the task of steering free of collisions with static and dynamic obstacles, while following a global plan to navigate towards a goal location. Static obstacles can be avoided using traditional planning algorithms whereas dynamic obstacles pose a tough challenge. The velocity obstacle [3] is a geometric representation of all velocities that will eventually result in a collision given that the dynamic obstacle maintains the observed velocity. In a multi-mobile robot scenario, robots can not merely be regarded as dynamic obstacles. Each robot is a pro-active agent, taking actions to avoid collisions. Neglecting this fact might lead to oscillations and thus highly inefficient trajectories or even collisions.

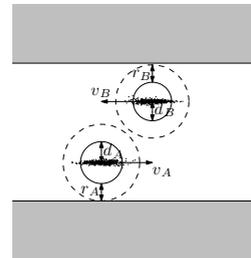
To construct velocity obstacles, relative position, current velocity and shape of all other robots are required. Many approaches avoid on-board sensing of this information using global positioning via an overhead tracking camera or assume perfect sensing, i.e. are purely simulation based. This limits the possibilities for application greatly. *Collision avoidance under localization uncertainty* (CALU) [5] combines the velocity obstacle approach with on-board localization. CALU provides a solution that is situated in-between centralized motion planning and communication-free individual navigation. While actions are computed independently for each robot, information about position and velocity is shared using local inter-robot communication. This keeps the communication overhead limited while avoiding problems like robot-robot detection. Uncertainty in localization is addressed by inflating the robots' radii according to the particle distribution of adaptive monte carlo localization (AMCL) [4].

## 2 Problem description and approach

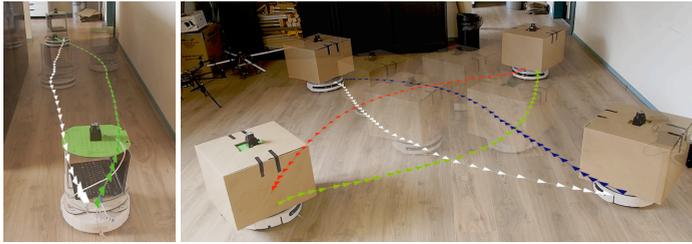
While CALU effectively alleviates the need for global positioning by using decentralized localization, some problems prevail. Suboptimal behavior is encountered when (a) the footprint of the robot is not efficiently approximated by a disk; and (b) the pose belief distribution of AMCL is not circular but elongated along one axis (typically observed in long corridors). In both situations, the resulting VOs vastly overestimate the unsafe velocity regions. Hence, this conservative approximation might lead to a suboptimal - or no solution at all. The corridor example, as presented in Figure 1, shows the shortcomings of this approach.

## 3 Convex outline collision avoidance under localization uncertainty

The key difference between CALU and COCALU is to use the shape of the particle cloud instead of using a circumscribed circle. In this approach, we approximate the shape of the particle filter by a convex hull. However, using the convex hull of all particles can result in large overestimations, since outliers inflate



**Figure 1:** The corridor problem: Approximating the localization uncertainty (and the footprint) with circumscribed circles, vastly overestimates the true sizes, such that the robots do not fit next to each other.



**Figure 2:** COCALU with two turtlebots in a corridor (left) and with four quadratic shaped turtlebots (right).

the resulting convex hull immensely. As a solution to this problem, we use *convex hull peeling*, which is also known as *onion peeling* [1], in combination with an error bound  $\epsilon$ . Onion peeling can be intuitively explained as removing all points on the convex hull, and to calculate a new convex hull of the remaining points. This process can be repeated iteratively until the remaining points are less than three.

COCALU finds the area for which the probability of the robot being located in is greater than the error bound  $1 - \epsilon$ . In order to find the specific convex hull that defines this area, we propose an iterative process: As long as the sum of the weights of the removed samples does not exceed the error bound, we create the convex hull of all (remaining) particle samples. Afterwards, we sum up all the weights of the particles located on the convex hull and add this weight to the previously computed sum. If the cumulative sum does not exceed the error bound, all the particles that define the current convex hull will be removed from the particle set and the process is repeated. When the convex hull is found, we calculate the Minkowski sum of the robot's footprint and the convex hull. The convex hull of the Minkowski sum is then used as new footprint of the robot.

## 4 Demonstration

We demonstrate our approach in simulation and a real-world setting. In simulation, robots are positioned on a circle and the goals located on the antipodal positions, i.e. each robot's shortest path is through the center of the circle. For experiments and detailed results of the proposed system, we refer to [2].

In addition to simulation runs, we present our approach on up to four differential drive Turtlebots<sup>1</sup>. The robots are based on the iRobots Roomba platform and have a diameter of 33.5 cm. In addition to the usual sensors, they are equipped with a Hokuyo URG laser-range finder to enable better localization in large spaces. All computation is performed on-board on a Intel i3 380UM 1.3GHz dual core CPU notebook. Communication between the robots is realized via a 2.4 GHz WiFi link.

We have tested a realistic setting of two robots in a narrow corridor. Each robot has to maneuver to the other side of the corridor; thus having to pass the other robot. Figure 2 (left) shows the setup and the resulting paths using COCALU. The second test uses four turtlebots with altered shapes (i.e. squared boxes) and uses a similar setup as in the simulation test. Figure 2 (right) shows the resulting paths using the quadratic shaped turtlebots.

## References

- [1] Bernard Chazelle. On the convex layers of a planar set. *IEEE Transactions on Information Theory*, 31(4):509–517, 1985.
- [2] Daniel Claes, Daniel Hennes, Karl Tuyls, and Wim Meeussen. Collision avoidance under bounded localization uncertainty. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Algarve, Portugal, October 2012.
- [3] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *Robotics Research*, 17:760–772, 1998.
- [4] Dieter Fox. Adapting the sample size in particle filters through kld-sampling. *Robotics Research*, 22, 2003.
- [5] Daniel Hennes, Daniel Claes, Karl Tuyls, and Wim Meeussen. Multi-robot collision avoidance with localization uncertainty. In *Int. Conf. on Autonomous Agents and Multiagent Systems*, 2012.

<sup>1</sup>For more information see: <http://turtlebot.com>

# Kamala in the Cloud: Bottom-up Knowledge Modeling Using the Topic Maps Data model

Fiemke Griffioen    Gabriel Hopmans    Peter-Paul Kruijssen    Quintin Siebers

*Morpheus Kennistechnologie BV, PO Box 69. 3500 CD Utrecht, The Netherlands*

## 1 Topic Maps

Kamala in the Cloud [1] is a platform developed by Morpheus Kennistechnologie (Knowledge Engineering) to effectively model domain knowledge in the cloud. Kamala is designed to provide a bottom-up method of modeling domain knowledge based on the Topic Maps standard. Topic Maps, defined in ISO/IEC 13250 [3], is a datamodel for the semantic structuring of linked networks. It is a technology that is used for knowledge representation and knowledge management.

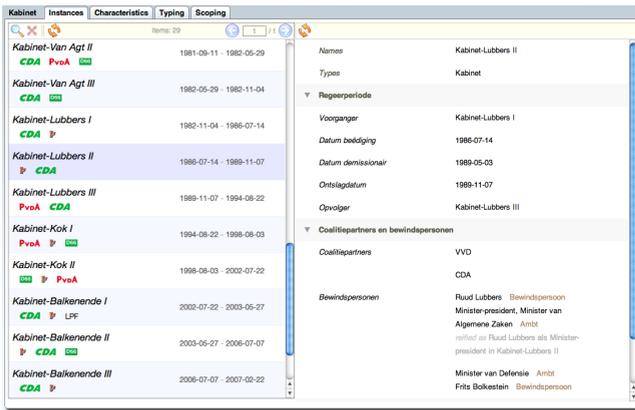
The basic building blocks of a topic map are topics, occurrences and associations. Topics are objects within a topic map that represent a subject of a specific type. For instance Beatrix is a topic of type person and the Netherlands is a topic of type country. Occurrences relate topics to the information they are relevant to. This can be in the form of URI's, similar to page numbers in a back-of-book index: they indicate where information about the subject can be found in the book. The information can also be entered directly, phone numbers, addresses and dates for instance. Associations are relations between topics, for example Beatrix is the queen of the Netherlands, or Sacha de Boer works for the NOS in the function of newsreader. Occurrences and associations are typed, allowing any kind of information resource or relationship to be defined.

## 2 Kamala in the Cloud

The Topic Maps datamodel forms the basis of the application Kamala in the Cloud. In Kamala, a knowledge model can be built without requiring extensive knowledge of Topic Maps and by using a bottom-up methodology. After logging in to the Kamala server, the domain modeling process can start by entering concrete domain data. Later in the process, the ways in which the data are interrelated (the meta-level or ontology) is modeled. This is the opposite approach from the standard top-down method starting with developing a full domain model before the first concrete data can be entered. The top-down method implicitly assumes that the user has a complete overview on the entire domain and is capable of conceptualizing this down to the smallest details. With the bottom-up method, the user can enter data, observe patterns and contribute to the knowledge model on the fly. Top-down modeling is possible in Kamala, however, in this demonstration the focus is on the bottom-up method.

The following features are included in Kamala in the Cloud: 1) availability of the complete data model of Topic Maps standard, 2) navigation based on ontological structures, 3) search topics based on naming, 4) sharing topic maps with other users (optionally read-only), 5) importing and exporting topic maps to the standard formats XTM, TMXML, LTM, CXTM, etc., 6) querying topic maps with the TOLOG [2] or TMQL query languages, 7) storing queries for reporting purposes, 8) validation of topic maps, so that gaps in the knowledge model can be traced, 9) generating statistics.

Kamala is a cloud application, which means no installation is needed. The platform only requires the availability of a standard browser. Figure 1a shows a screenshot of a topic list in the topic map about the Dutch government. Typical customers for Kamala are organizations that deal with large amounts of unstructured or unlinked data. Currently, Kamala is being used by, amongst others, the Dutch tax office, a police department and the Dutch House of Representatives.

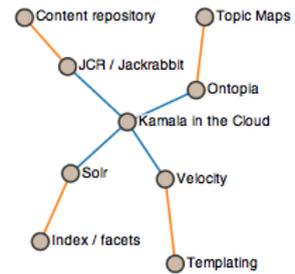


Instances	Characteristics	Typing	Scoping
Kabinet-Van Agt II	1981-09-11 - 1982-05-29	CDA PvdA	
Kabinet-Van Agt III	1982-05-29 - 1982-11-04	CDA	
Kabinet-Lubbers I	1982-11-04 - 1986-07-14	CDA	
Kabinet-Lubbers II	1986-07-14 - 1989-11-07	CDA	
Kabinet-Lubbers III	1989-11-07 - 1994-08-22	PvdA CDA	
Kabinet-Kok I	1994-08-22 - 1998-08-03	PvdA	
Kabinet-Kok II	1998-08-03 - 2002-07-22	CDA PvdA	
Kabinet-Balkenende I	2002-07-22 - 2003-05-27	CDA LFP	
Kabinet-Balkenende II	2003-05-27 - 2006-07-07	CDA	
Kabinet-Balkenende III	2006-07-07 - 2007-02-02	CDA	

Names	Kabinet-Lubbers II
Types	Kabinet
Regeerperiode	
Voorangar	Kabinet-Lubbers I
Datum beëdiging	1986-07-14
Datum demissionair	1989-05-03
Ontslagdatum	1989-11-07
Opvolger	Kabinet-Lubbers III
Coölitiepartners en bewindspersonen	
Coölitiepartners	
VVD	
Bewindspersonen	
Raai Lubbers	Bewindsperson
Minister-president, Minister van	
Algemene Zaken	Ambt
reflected as Raai Lubbers als Minister-president in Kabinet-Lubbers II	
Minister van Defensie	Ambt
Frits Bolkestein	Bewindsperson

(a) Topics can easily be navigated using the topic list



(b) Kamala in the Cloud: the integration of open source products

Figure 1

Several open source products are used by Kamala for different functionalities, see Figure 1b. First of all, it is based on Ontopia, a Topic Maps framework. With Solr, the knowledge model can be indexed with keywords as well as user-defined facets. Velocity is a templating engine, which allows users to create their own look and feel for their knowledge model by defining HTML and java-based templates. These templates and other uploaded content is stored using Jackrabbit, an implementation of the Java Content Repository standard.

There are a number of modules available to expand Kamala's core functionality. The above mentioned Velocity templating (e.g. the political party logos in Figure 1a) and facet-based classification with the Solr search platform are available as modules. Furthermore, the geo-module allows for geotagging topics placing them onto a map. A visualization tool is being developed to view topics and the relationships graphically. Figure 1b is an example of such a visualization.

### 3 The Demonstration

The requirements for participating in the demo are a browser and an Internet connection. After a short introduction, the audience can log on to the Kamala website and collaborate on a topic map about the BNAIC conference. Finally, we will show some additional features and modules of Kamala in the Cloud. The demo will take approximately 30 minutes.

### References

- [1] <http://kamala.mssm.nl>.
- [2] Lars Marius Garshol. TOLOG – A Topic Maps Query Language. In *Charting the Topic Maps Research and Applications Landscape*, volume 3873 of *Lecture Notes in Computer Science*, pages 183–196. Springer, 2006.
- [3] ISO/IEC IS 13250-2:2006. *Information Technology - Document Description and Processing Languages - Topic Maps - Data Model*. International Organization for Standardization, Geneva, Switzerland.

# Demonstration of eMate – stimulating behavior change via mobile phone

Michel Klein<sup>a</sup> Nataliya Mogles<sup>a</sup> Arlette van Wissen<sup>a</sup>

<sup>a</sup> Artificial Intelligence, VU University Amsterdam, the Netherlands  
De Boelelaan 1081, 1081 HV Amsterdam

## 1 The eMate system

### 1.1 Aims and functionality

eMate is an intelligent system that aims to support patients with Diabetes Mellitus type II, heart diseases or HIV in adhering to their therapy. The therapy consists of lifestyle advice and/or precise instructions for medication intake. Although the system has been implemented for those three diseases, the system can be used for all scenarios in which behavior change is important.

Research has shown that a ‘cooperative assistant’ – i.e., with a coaching character, able to explain and educate, and expecting high participation of the user – is more effective than a ‘direct assistant’ – i.e., with an instructing character with brief reporting and low expectations on participation [1]. The eMate system therefore operates as a coach, using both a mobile phone and a website to interact with the user. Via the website, the user can get an overview of his progress on three different domains: medication intake, physical exercise, and healthy food intake. An overview shows the extent to which the user has reached his/her goals in the past week, which is represented as a percentage and an iconic thumb. **Error! Reference source not found.** shows the overview page of the website. A mobile phone application for the Android platform has been developed that can pose questions and send messages to a user. Figure 1 gives an impression of one specific screen of the mobile system.

The system has been developed in the context of a ZonMW funded project “Intelligent Monitoring and support of chronic patients”. It has been developed by researchers of the VU University Amsterdam in collaboration with Evalan BV, a company that develops innovative telemetry solutions for health care.

### 1.2 Model-based reasoning

The kernel of the system is a computational model of behavior change, based on theoretical frameworks [2]. The model is used to analyze the state of the patient with respect to his/her behavior change goals. It does so by investigating via simple questions which of the factors that influence behavior change are probably the most problematic for this patient. This mechanism is called model-based diagnosis [3]. These factors are then targeted with specific messages and interventions. For this purpose the model is represented into a rule-based format that allows for backward reasoning over the psychological factors in the model.



Figure 1. Screen of the mobile application



Figure 2. Overview page of the eMate website.

## 2 Demo scenario

The demonstration consists of a time-lapse walkthrough of the interaction of a patient with the system, illustrated by a visualization of the reasoning process that takes place within the system. It will be shown how the patient initializes the system via a web-based questionnaire, and how this will result in specific values for the factors in the computational model of behavior change. Using a graphical representation of the computational model, it will be shown how the model-based diagnosis process will lead to hypothesis about the psychological causes of the behavior, and how this will lead to questions to the patient via the mobile application. These questions will be answered on the phone, and again it will be illustrated how these answers are processed and fed into the computational model. After a number of interactions, this will result in specific messages to the patient that are delivered via the mobile phone. In this way, the demonstration mimics the interaction that a patient has with the system during several weeks in a few minutes.

The total duration of the demonstration is 15 minutes. The demonstration requires a computer with internet access and a large screen (for the illustration of the reasoning process), a laptop for controlling the system, and a mobile phone for the end-user interaction with the system. The latter two components will be provided by the demonstrators.

## References

- [1] Henkemans, O.A.B., Rogers, W.A., Fisk, A.D., Neerincx, M.A., Lindenberg, J., van der Mast, C.A.P.G.: Usability of an adaptive computer assistant that improves self-care and health literacy of older adults. *Methods of Inf. in Medicine* 47, 82–88 (2007).
- [2] Klein, M. C.A., Mogles, N. and Wissen, A. van.: Why won't you do what's good for you? Using intelligent support for behavior change. In Albert Salah and Bruno Lepri, editors, *Human Behavior Understanding*, volume 7065 of LNCS, pages 104–115. Springer Berlin / Heidelberg, 2011.
- [3] Davis, R.: Diagnostic reasoning based on structure and behavior. *Artificial Intelligence* 24(1-3), 347–410(1984)

# Scheduling with precedence constraint posting

Michel Wilson<sup>a</sup>

Cees Witteveen<sup>a</sup>

Bob Huisman<sup>b</sup>

<sup>a</sup>*Delft University of Technology, Dept of Software Technology*

<sup>b</sup>*Nedtrain, Fleet Services – Maintenance Development*

## Abstract

In this demonstration, we show a software tool designed to interact with a scheduling algorithm. The algorithm solves the Resource-Constrained Multi-Project Scheduling Problem [3], which consists of projects with a release and due date, where the projects are composed of tasks, using one or more resources with a limited capacity. The tool serves a dual purpose: it is used to demonstrate the possibilities of using scheduling algorithms as a decision support system to end-users (i.e., schedulers), and it is used to gain insight into how scheduling algorithms solve different problem instances, and how different settings can influence the solutions to a particular instance.

NedTrain is a subsidiary of NS Group, the Dutch national passenger railway operator, which is responsible for maintenance of the rolling stock. Efficiently scheduling maintenance operations in their workshops is seen as an important topic in their current research program, and research is being done on scheduling methods which can cope well with the dynamic and uncertain nature of maintenance operations scheduling. Within this program, a demonstrator tool has been developed together with a group of bachelor students [1], with the dual purpose of demonstrating the capabilities and possibilities of the algorithms developed, and of giving insight into the inner workings and differences between scheduling algorithms.

## 1 Problem definition and algorithm outline

The problem of scheduling maintenance operations on trains in a workshop can be generalized to the Resource-Constrained Multi-Project Scheduling Problem (RCMPSP). This problem consists of the following components:

- A set  $P$  with  $m$  projects, where each project  $p_i \in P$  has a release date  $rel_i \in \mathbb{N}$  and due date  $due_i \in \mathbb{N}$ .
- Each project  $p_i \in P$  has a set  $T_i$  of  $n_i$  tasks, where each task  $t_{i,j} \in T_i$  has a duration  $d_{i,j} \in \mathbb{N}^+$ .
- A precedence relation  $\prec$  inducing a partial order on  $\bigcup_i T_i$ ; the intuitive interpretation is that  $t_{i,j} \prec t_{p,q}$  means that  $t_{i,j}$  has to finish before  $t_{p,q}$  can start.
- A set  $R$  of  $\ell$  renewable resources, where each resource  $r_k \in R$  has an integer capacity  $cap(r_k) \leq 1$ .
- For each  $r_k \in R$  and  $t_{i,j}$ ,  $req(t_{i,j}, r_k)$  specifies the amount task  $t_{i,j}$  needs of resource  $r_k$  in order to be executed.

The goal is to find a solution to the problem, in the form of a start time assignment  $\sigma : \bigcup_i T_i \rightarrow \mathbb{N}$ , such that all the constraints are specified.

The method employed to find solutions is based on precedence constraint posting [2, 4] with task grouping [5], which iteratively executes the following steps:

1. A provisional solution  $\sigma$  is made by computing the earliest start times for all tasks using the current set of precedence constraints  $\prec$ .
2. Using this provisional solution, the resource usage profile is computed over time for each resource. If usage is below capacity for each resource at each time point, the current solution is valid. If not, we select the resource peak which violates the capacity by the largest amount.

3. Two (partially) concurrent tasks  $t_{i,j}$  and  $t_{p,q}$  contributing to that resource peak are selected.
4. To lower the resource peak, the two selected tasks are forced to execute sequentially, by either posting a constraint of the form  $t_{i,j} \prec t_{p,q}$  or  $t_{p,q} \prec t_{i,j}$ , or by grouping  $t_{i,j}$  and  $t_{p,q}$  together, such that their exact order is determined at execution time.

## 2 Software features

The developed software contains all the basic features to interact with problem instances. All aspects of an instance can be modified using a visual interface, such as resource usage of a task, and deadlines of projects. Additionally, more advanced features are available such as inserting projects based on templates in an existing instance, and adding a temporary decrease in resource capacity to simulate, for example, a machine outage due to maintenance.

The solver is a separate program, with a well-defined interface, such that multiple solvers can be used in the future. To run the solver, a parameter screen is available, to easily specify whether grouping should be enabled, and with which threshold. A solved instance is displayed similarly to an unsolved one.

It is possible to trace through the steps taken by the solver to reach the solution, whereby the provisional solutions are displayed, and the constraint or task group added to reach this solution is highlighted. If no solution can be found, the resource peak or project due date causing the problem is highlighted. The user can then modify some aspects of the instance, for example the capacity of a resource, and try to solve the instance again.

## 3 Conclusion

The demonstrated software contains features useful for both planners and researchers. Planners can get a better idea of the ways in which software can let them interact with their planning problems and with algorithms solving these problems, and researchers can get a better grasp on which steps an algorithm goes through to solve a particular instances, and where the problem spots are in instances which are not solvable.

## References

- [1] Erik Ammerlaan, Jan Elffers, Erwin Walraven, and Wilco Wisse. Visualisatie en ondersteuning bij planning en scheduling, 2012. Bachelor thesis.
- [2] Amedeo Cesta, Angelo Oddi, and Stephen F Smith. Profile-based algorithms to solve multiple capacitated metric scheduling problems. In *Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems*, pages 214–223, 1998.
- [3] Sönke Hartmann and Dirk Briskorn. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1 – 14, 2010.
- [4] Angelo Oddi, Amedeo Cesta, Nicola Policella, and Stephen F. Smith. Combining variants of iterative flattening search. *Engineering Applications of Artificial Intelligence*, 21(5):683–690, August 2008.
- [5] Michel Wilson, Cees Witteveen, and Bob Huisman. Enhancing predictability of schedules by task grouping. In *20th European Conference on Artificial Intelligence*, 2012.

# List of authors

## A

Steven Aanen	269
Stéphane Airiau	3
Huib Aldewereld	297
Laura Antanas	75
Reyhan Aydoğan	327

## B

Tim Baarslag	271
Hendrik Baier	273
Henri Bal	325
Sander Beckers	11
Wouter Beek	275
Antonis Bikakis	83
Yngvi Björnsson	323
Hendrik Blockeel	321
Janneke Bolt	19, 27
Richard Booth	277
Antal van den Bosch	147
Peter Bosman	279, 305
Tibor Bosse	281
Haitham Bou Ammar	51, 59, 283
Paul Bouman	285
Bert Bredeweg	275
Guy Van Den Broeck	317
Maurice Bruynooghe	329
Tim Brys	287
Harm Buisman	289
Wouter Bulten	35

## C

Patrice Caire	83
Martin Caminada	43, 343
Mohammad Chami	51
Siqi Chen	59
Daniel Claes	345
Martine De Cock	287
Silvano Colombo Tosatto	67

## D

Mehdi Dastani	234, 337
Jesse Davis	321
Marc Denecker	329
Kathrin Dentler	291
Michael Derde	75
Virginia Dignum	258, 297, 339
Alex Dirkzwager	271
Kurt Driessens	283

## E

Vasileios Eftymiou	83
Armin Elbers	107
Ulle Endriss	91

## F

Mike Farjam	99
Eduardo Fermé	277
Daan Fierens	321

Flavius Frasincar	269, 293
-------------------	----------

## G

Linda van der Gaag	27, 107, 242
Mattijs Ghijsen	115
Toon Goedeme	11
Guido Governatori	67
Fiemke Griffioen	347
Christophe Guéret	291
Fabian Guiza Grandas	75

## H

Frank van Harmelen	325
Willem Haselager	35, 99
Yann-Michaël De Hauwere	178, 287
Bram Havers	170
Emile Hendriks	307
Mark Hendrikx	271
Daniel Hennes	345
Maurice Hermans	123
Koen Hindriks	271, 327
Frederik Hogenboom	293
Mark Hoogendoorn	281
Johan Hoorn	315
Gabriel Hopmans	347
Bob Huisman	333, 351
Yannick van den Hurk	162
Evelien van der Hurk	285

## I

Wouter IJntema	293
----------------	-----

## J

Wojtek Jamroga	295
Wouter Jansweijer	115
Jie Jiang	297
Catholijn Jonker	258, 271, 327, 339
Bart Joosten	299

## K

Leslie Kaelbling	311
Pierre Kelsen	67
Marwane El Kharbili	67
Jeesun Kim	299
Michel Klein	281, 301, 349
Tomas Klos	194
Sébastien Konieczny	277
Walter Kosters	139
Spyros Kotoulas	325
Emiel Krahrmer	299
Leo Kroon	285
Peter-Paul Kruijsen	347
Florian Kunneman	147
Johan Kwisthout	131

## L

Ting Li	285
Willie Loeffen	107

Marco Loog	303
Milan Lovric	285
Menno Luiten	139
Hoang Ngoc Luong	305

## M

Qin Ma	67
Jason Maassen	325
Laurens van der Maaten	307
Freek Maes	155
Bernard Manderick	250
Ruud Mattheij	162
Matthijs Melissen	295
John-Jules Meyer	337
Gabriel Mititelu	170
Kristof Van Moffaert	178
Nataliya Mogles	301, 349
Adriaan ter Mors	186

## N

Ali Nakisaee	210
Lennart Nederstigt	269
Frits de Nijs	194
Pim Nijssen	309
Siegfried Nijssen	317
Ann Nowé	178, 287

## O

Frans Oliehoek	311, 319
----------------	----------

## P

Tom Pepels	313
Ramon Pino Pérez	277
Mikołaj Podlaszewski	343
Matthijs Pontier	315
Eric Postma	162, 289, 299

## Q

Wenzhao Qiao	202
--------------	-----

## R

Carin Rademaker	242
Luc De Raedt	75
Jan Ramon	331
Bijan Ranjbar-Sahraei	210
Joris Renkens	317
Nico Roos	202

## S

Gorik De Samblanx	11
Frederik Schadd	123
Joris Scharpff	218
Hermi Schijf	107
Stefan Schlobach	291
Johannes Scholtes	155
Martijn Schut	291
Quintin Siebers	347
Robert-Jan Sips	170
Floris De Smedt	11

Matthijs Spaan	218, 319
Ida Sprinkhuizen-Kuyper	35, 99
Pieter Spronck	162
Lars Struyf	11
Marc Swerts	299
Zoltán Szlávik	170

## T

Nima Taghipour	321
Mandy Tak	323
Frank Takes	139
Yao-Hua Tan	297
David M.J. Tax	226
Matthew E. Taylor	283
Bas Testerink	234
Dirk Thierens	279
Leon van der Torre	67, 337
Jan Treur	281
Karl Tuyls	51, 59, 283, 345

## U

Jacopo Urbani	325
---------------	-----

## V

Damir Vadic	269
Joost Vennekens	11, 329
Peter Vervest	285
Sicco Verwer	335
Srdjan Vesic	43
Wietske Visser	327
Hanne Vlaeminck	329
Holger Voos	51
Peter Vrancx	178

## W

Natalie van der Wal	281
Yuyi Wang	331
Mathijs de Weerd	218
Gerhard Weiss	51, 59, 210, 283
Bob Wielinga	115
Daan Wilmer	194
Michel Wilson	333, 351
Mark Winands	273, 309, 313, 323
Arlette van Wissen	281, 301, 349
Cees Witteveen	186, 333, 351
Stefan Witwicki	311
Steven Woudenberg	242

## Y

Saba Yahyaa	250
-------------	-----

## Z

Yingqian Zhang	335
Pouyan Ziafati	337
Janneke van der Zwaan	258, 339